AUTOMATED BENCHMARKING IN CONTENT-BASED IMAGE RETRIEVAL

Henning Müller, Wolfgang Müller,
Stéphane Marchand-Maillet and Thierry Pun

Computer Vision Group, University of Geneva
Geneva, Switzerland

David McG. Squire

CSSE, Monash University,
Melbourne, Australia

ABSTRACT

Benchmarking has always been a crucial problem in content-based image retrieval (CBIR). A key issue is the lack of a common access method to retrieval systems, such as SQL for relational databases. The Multimedia Retrieval Mark-up Language (MRML) solves this problem by standardizing access to CBIR systems (CBIRSs). Other difficult problems are also shortly addressed, such as obtaining relevance judgments and choosing a database for performance comparison. In this article we present a fully automated benchmark for CBIRSs based on MRML, which can be adapted to any image database and almost any kind of relevance judgment. The test evaluates the performance of positive and negative relevance feedback, which can be generated automatically from the relevance judgments. To illustrate our purpose, a freely available, non-copyright image collection is used to evaluate our CBIRS, *Viper*. All scripts described here are also freely available for download.

1. INTRODUCTION

An increasing amount of research in the areas of computer vision and pattern recognition deals with the field of content-based image retrieval (CBIR). Many techniques have been developed for specialized fields and new features are developed regularly. The biggest problem with all this is that it remains basically impossible to compare the effectiveness and/or efficiency of the retrieval techniques and image features. In making such a comparison, it is essential to have a means of comparing several systems on the same grounds. Only by using such a performance measuring tool can systems be compared and the better techniques identified.

The basis for such a benchmark must be a set of common image or multimedia databases. At present, the most commonly used images come from the Corel Photo CDs, each of which contains 100 broadly similar images (`http://www.corel.com/`, *e.g.*[1]). Unfortunately these images are copyrighted, and are not free of charge. Most research groups use only a subset of the entire collection. This can result in a collection containing several highly dissimilar image groups, with relatively high within-group similarity, leading to great apparent improvements in performance. A good candidate for a standard collection could be the images and videos from MPEG-7 [6]. Unfortunately they may not be shown on the web, and the collection is expensive. Alternatively, CBIR researchers could develop their own collection. Such a project is underway at the University of Washington in Seattle (see `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`). This collection is freely available and is not copyrighted. It offers annotated photographs of different regions and topics. Currently it is small ($\sim$ 900 images), but several groups are contributing to enlarge the data set. This collection will be used to demonstrate the benchmark with the *Viper* CBIR system (CBIRS).

Obtaining relevance judgments for the benchmark queries presents another problem. Ideally real users should be involved [8], but initially pre-defined collection categories may be used. More about categorization of images can be read in [5].

Many CBIR performance measures have been proposed. An overview is given in [8], and a more formal way to develop measures without stating any precise measures is given in in [11]. In the automated performance benchmark descried here, we use several measures inspired by those used at the TREC conferences [14] (see `http://trec.nist.gov/`) in text retrieval. It is possible that TREC will integrate images into their evaluation procedure, as was done earlier with other areas of information retrieval, such as interactive systems [12] and videos.

Maybe the biggest problem in automatically benchmarking CBIRSs is the lack of a common access method. The advent of the MRML [10] has solved this problem. MRML standardizes CBIRS access. It allows a client to log onto a database and ask for the available image collections as well as to select a certain similarity measure, and to perform queries using positive and negative examples. With such a communication protocol the fully automated evaluation of CBIRSs is possible.

2. BENCHMARKING IN IMAGE RETRIEVAL

There have been few publications about benchmarking in CBIR to date. Subfields, such as the development of performance measures have been discussed [8, 11]. In [2] one measure to compare two systems is shown. This evaluation of CBIRSs was the goal of EU project MIRA (Evaluation Frameworks for Interactive Multimedia Information Retrieval) (see `http://www.dcs.gla.ac.uk/mira/`). Several web pages give comparisons of systems based on a number of key CBIR features the systems offer (see `http://compass.itc.it/`), such as feedback methods or the number of images displayed on screen. To our knowledge, no quantitative evaluation of the performance of several systems has yet been made. In [15] the performance of three systems is compared. This is at least a beginning, since several groups agreed to participate. No quantitative performance measures are given—a few example queries and the system responses are shown—which is inadequate.

This work was supported in part by the Swiss National Foundation for Scientific Research (grant no. 2000-052426.97).

The most profound study so far has been started by the Institute for Image Data Research at the University of Northumbria at Newcastle, in a project sponsored by the British Joint Information Systems Committee (JISC). They have downloaded and compared many CBIRSs. The first report [3] gives a very good summary of the different techniques used, but the performance comparison of some of the downloaded systems in [13] is unfortunately not very profound.

By far the most promising approach is the *Benchathlon* (see `http://www.benchathlon.net/`), which is trying to start a regular benchmark event where CBIRSs can be compared. The structure of the benchmark is described in [4].

For image browsing systems, such as `PicHunter` or *Tracking-Viper*, a benchmark which attempts to simulate user behavior by using an extensively annotated collection has been proposed [9].

3. MULTIMEDIA RETRIEVAL MARK-UP LANGUAGE

MRML (see `http://mrml.net/`, [10]) is an XML-based communication protocol for content-based query, which was developed to separate the query interface from the actual query engine. It was specially developed for CBIR and thus contains tags for query by positive and negative examples.

An MRML server listens on a port for messages. When connecting, the MRML client requests the basic server properties. The MRML message looks like this:

```
<mrml> <get-server-properties /> </mrml>
```

The server then informs the client of its capabilities:

```
<mrml> <server-properties>
    <vi-collectionlist>
      <vi-collection
        name="WashingtonGroundtruth" />
    </vi-collectionlist>
</server-properties></mrml>
```

Using similar simple messages, the client can request a list of the collections available on the server, together with descriptions of the ways in which they can be queried. The client can open a session on the server, and configure it according to the needs of its user (interactive client) or its own needs (*e.g.*benchmark test).

A basic query consists of a list of images and their corresponding relevance levels, assigned by the user. In the following example, the user has marked two images: `1.jpg` positive and `2.jpg` negative. All images are referred to by their URLs.

```
<mrml session-id="1" transaction-id="44">
<query-step session-id="1"
 resultsize="30"
 <user-relevance-list>
  <user-relevance-element
   image-location="http://viper.unige.ch/1.jpg"
   user-relevance="1"/>
  <user-relevance-element
   image-location="http://viper.unige.ch/2.jpg"
   user-relevance="-1"/>
 </user-relevance-list>
</query-step> </mrml>
```

The server will return the retrieval result as a list of image URLs, ordered by their relevance to the query.

The key to the successful extension of MRML is graceful degradation. This means that servers and clients which do not recognize an XML element or attribute encountered in an MRML text should ignore its contents completely. To avoid conflicts between extensions of MRML, it is planned to maintain a central database for the registration and documentation of MRML extensions.

4. THE AUTOMATED BENCHMARK

This section describes in detail the techniques used in the automated benchmark, as well as ways to modify it to adapt it to other systems and databases, or to add extra performance measures.

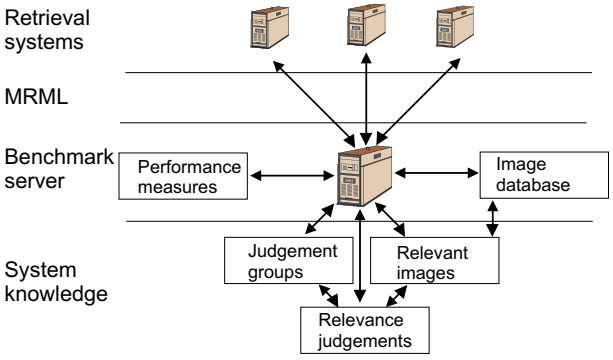


Figure 1: Structure of the automated benchmark.

Figure 1 shows the basic structure of the benchmark. MRML serves as the communication layer between the evaluated systems and the benchmark server. The image database and the performance measures are known to all the systems. The relevance judgments, however, should not be known since the responses can easily be optimized with this knowledge. Initially, the relevance judgments will also be made available as they are the database groups.

4.1. Performance measures

All the performance measures described in [8] are used, in order to be similar to TREC. They are:

- $Rank_1$, $\overline{Rank}$ and $\widetilde{Rank}$: rank at which first relevant image is retrieved, average rank, normalized average rank of relevant images (see below and Eq. 1);
- $P(20)$, $P(50)$ and $P(N_R)$: *precision* after 20, 50 and N_R images are retrieved. N_R is the number of relevant images;
- $R_P(.5)$ and $R(100)$: *recall* at *precision* .5 and after 100 images are retrieved;
- *precision/recall* graph.

A simple average rank is difficult to interpret since it depends on both the collection size N and the number of relevant images N_R for a given query. Consequently, we normalize by N and N_R and propose the *normalized average rank*, $\widetilde{Rank}$:

$$\widetilde{Rank} = \frac{1}{NN_R}\left(\sum_{i=1}^{N_R} R_i - \frac{N_R(N_R-1)}{2}\right) \qquad (1)$$

where R_i is the rank at which the ith relevant image is retrieved. This measure is 0 for perfect performance, and approaches 1 as performance worsens.

Additional measures can be added at any time. Our experience so far shows that these measures, especially the precision/recall graphs, give a good overview of the performance of a CBIRS.

4.2. Benchmarking Software

The benchmark is implemented as a Perl object. The only parameters that need to be set to run the benchmark are the hostname and port of the query engine to be evaluated. For a new image database, the location of the query relevance judgments, relevance groups and relevant images must be set.

Figure 2 shows the basic communication when running the benchmark. First, the server to be evaluated must be configured.

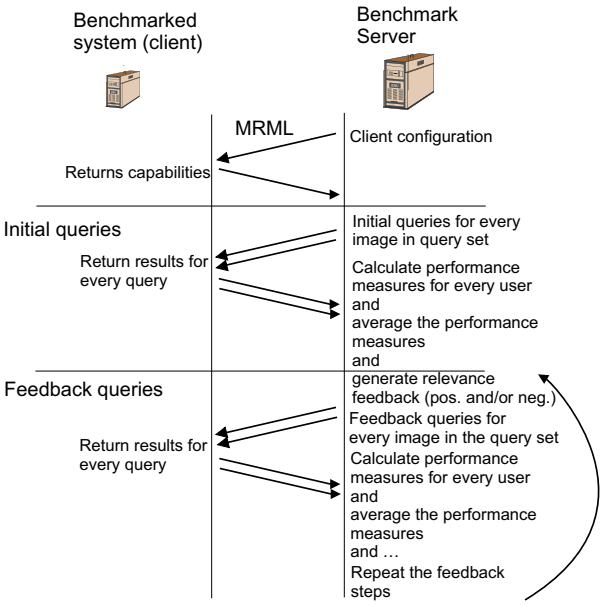


Figure 2: Communication flow for the automated benchmark.

Then, a query is executed for each query image. The returned results are evaluated on the basis of the relevance judgments. Positive and negative relevance feedback (RF) is simulated, based on the relevance judgments. Using the simulated RF, another series of queries is executed and the results are evaluated. This feedback step can be repeated to refine the query, in order to evaluate the effectiveness of RF for the evaluated system.

4.2.1. Reading the base data for the evaluation

Three inter-related data sets are required for system initialization:

- the list of images for the first query step,
- the list of relevance judges (one for each person who made judgments, or one only if the database grouping is used),
- a relevance judgment file for every image/relevance judge combination containing a list of all images regarded as relevant for a certain query image.

These files are currently in plain text, but it is planned to use XML.

4.2.2. Generation of relevance feedback

RF is generated from the relevance judgment files and the system response. We assume that the user would mark all relevant images positively and all non-relevant images negatively. We also need to assume the number of images the user would view. We choose 20 as a typical number of images displayed by a CBIRS. Thus, all images from the first 20 images of the system response which are in the relevant set for the query image are fed back positively and all those not in the relevant set are fed back negatively. See [7] for further details.

4.2.3. Evaluation

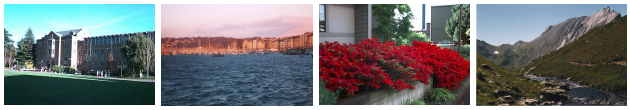
We perform an evaluation for each image/relevance judge combination. For the initial image and for each step of RF. The results are averaged over all queries, with the aim of obtaining robust and meaningful results.

4.3. Configuring the benchmark for other databases

It is very easy to configure the benchmark for a new database. It is only necessary to create the query image file, the relevance judges file, and, for each query image/relevance judge combination, a relevance judgment file. In the simplest case, when the database is organized into groups, one image from each group is used as a query, the relevance judges file has only one entry, and the database organization is used to construct the relevance judgment file.

5. EXAMPLE EVALUATION

In this section the results of the automated benchmark will be given based on the database of the University of Washington. The entire test can be downloaded at (see `http://viper.unige.ch/`). Figure 3 shows four of the query images of the database.



Campus Geneva Flowers Mountains

Figure 3: Sample images from the Washington test database.

Table 1 shows the results for the initial query and four steps of RF for the *Viper* system. The benchmark configuration was constructed from the database organization (see §4.3). The results shown are averaged over the 14 query images. There are 922 images in the database, a different number in each group.

Measure	no RF	RF 1	RF 2	RF 3	RF 4
N_R	65.14	65.14	65.14	65.14	65.14
t	1.23 s.	2.18 s.	2.49 s.	2.62 s.	2.70 s.
$Rank_1$	1.5	1	1	1	1
$R(P(.5))$	.3798	.5520	.6718	.6594	.7049
$\widetilde{Rank}$	176.44	152.28	116.13	107.04	104.37
$\widetilde{Rank}$	.1583	.1318	.0921	.0821	.0793
$P(20)$	.5392	.7357	.8642	.8892	.9107
$P(50)$	.4057	.5271	.6085	.6328	.6257
$P(N_R)$	.3883	.5256	.6138	.6640	.6553
$R(100)$	.4839	.6070	.6924	.7279	.7208

Table 1: Overview of the results for *Viper*

Figure 4 shows the average precision/recall graphs for the queries. This is the performance measure with the highest information content. The behavior of the system without RF and the

strong improvements with RF can easily be seen. The fourth RF step gives only a minor performance gain.

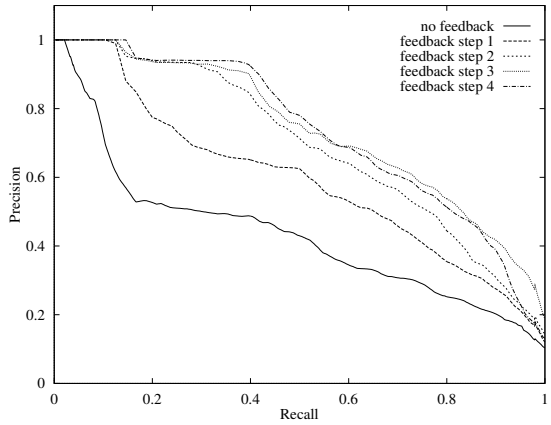


Figure 4: Precision/recall graphs without and with RF.

The performance measures in Table 1 are meant to complement the precision/recall graph. For a user, the precision of the images shown on screen is most important. We assume that the user looks at 20 to 50 images, so the precision at these points is very important. A significant improvement in each of the first three RF steps can be seen.

6. CONCLUSION

This paper presents a fully automated benchmark for CBIR which is completely based on freely available components. The goal is to standardize the evaluation of CBIRSs, and thus to make the quality of retrieval results comparable. The performance measures chosen are similar to those used in TREC, since this is the pre-eminent existing evaluation forum in information retrieval, and we wish to contribute to the establishment of a similar platform for CBIR.

We hope that other groups will make their image collections and relevance judgments available so it will really be possible to compare system performances fairly and quantitatively.

We want to encourage other research groups to use MRML as a communication protocol by making the benchmark available. A stable version of our program *Viper* is now called GIFT (GNU Image Finding Tool) and is available under a GNU license (see `http://www.gnu.org/software/gift/`).

There are still many problems to solve before a TREC-like benchmark can be performed in CBIR. For an objective benchmark the relevance judgments, as well as the image groups, should not be known, as this leaves room for manipulation. Any system can give a perfect response if the system knows which images need to be transmitted to achieve a perfect score. It is also necessary to have multiple real user judgments, since only with several judgments per query image can we show the ability of a system to adapt to the users' needs.

7. REFERENCES

[1] S. Belongie, C. Carson, H. Greenspan, et al. Color- and texture-based image segmentation using EM and its application to content-based image retrieval. In *Proceedings of the International Conference on Computer Vision (ICCV'98)*. Bombay, India, January 1998.

[2] A. Dimai. Assessment of effectiveness of content-based image retrieval systems. In D. P. Huijsmans and A. W. M. Smeulders, eds., *Third International Conference On Visual Information Systems (VISUAL'99)*, no. 1614 in Lecture Notes in Computer Science, pp. 525–532. Springer-Verlag, Amsterdam, The Netherlands, June 2–4 1999.

[3] J. Eakins and M. Graham. Content-based image retrieval. Tech. Rep. Technical Report JTAP-039, JISC Technology Application Program, 2000.

[4] N. J. Gunther and G. Beretta. A benchmark for image retrieval using distributed systems over the internet: Birds-i. Tech. rep., HP Labs, Palo Alto, Technical Report HPL-2000-162, San Jose, 2001.

[5] C. Jörgensen. Classifying images: Criteria for grouping as revealed in a sorting task. In *Proceedings of the 6th ASIS SIG/CR Classification research Workshop*, pp. 65–78. Chicago, Il, USA, October 1995.

[6] MPEG Requirements Group. MPEG-7: Context and objectives (version 10 Atlantic City). Doc. ISO/IEC JTC1/SC29/WG11, International Organisation for Standardisation, October 1998.

[7] H. Müller, W. Müller, S. Marchand-Maillet, et al. Strategies for positive and negative relevance feedback in image retrieval. In A. Sanfeliu, J. Villanueva, M. Vanrell, et al., eds., *Proceedings of the International Conference on Pattern Recognition (ICPR'2000)*, pp. 1043–1046. Barcelona, Spain, sep 3–8 2000.

[8] H. Müller, W. Müller, D. M. Squire, et al. Performance evaluation in content-based image retrieval: Overview and proposals. *Pattern Recognition Letters*, 2001.

[9] W. Müller, S. Marchand-Maillet, H. Müller, et al. Towards a fair benchmark for image browsers. In *SPIE Photonics East, Voice, Video, and Data Communications*. Boston, MA, USA, November 5–8 2000.

[10] W. Müller, H. Müller, S. Marchand-Maillet, et al. MRML: A communication protocol for content-based image retrieval. In *International Conference on Visual Information Systems (Visual 2000)*. Lyon, France, November 2–4 2000.

[11] A. D. Narasimhalu, M. S. Kankanhalli, and J. Wu. Benchmarking multimedia databases. *Multimedia Tools and Applications*, 4:333–356, 1997.

[12] P. Over. A review of Interactive TREC. In *MIRA workshop*. Dublin, Ireland, October 1998.

[13] C. C. Venters and M. Cooper. Content-based image retrieval. Tech. Rep. Technical Report JTAP-054, JISC Technology Application Program, 2000.

[14] E. M. Vorhees and D. Harmann. Overview of the seventh text retrieval conference (TREC-7). In *The Seventh Text Retrieval Conference*, pp. 1–23. Gaithersburg, MD, USA, November 1998.

[15] J. Ze Wang, G. Wiederhold, O. Firschein, et al. Wavelet-based image indexing techniques with partial sketch retrieval capability. In *Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries*, pp. 13–24. Washington D.C., May 1997.