An Open Framework for Distributed Multimedia Retrieval

Henning Müller Wolfgang Müller David McG. Squire* Zoran Pečenović†
Stéphane Marchand–Maillet Thierry Pun
Computer Vision Group, University of Geneva, 24 rue du Général Dufour,
CH - 1211 Geneva 4, Switzerland
*Computer Science and Software Engineering, Monash University
Melbourne, Australia
†Laboratoire de Communications Audio–Visuelles
Ecole Polytechnique Fédérale de Lausanne, Switzerland

Abstract

This article describes a framework for distributed multimedia retrieval which permits the connection of compliant user interfaces with a variety of multimedia retrieval engines via an open communication protocol, MRML (Multi Media Retrieval Markup Language). It allows the choice of image collection, feature set and query algorithm during run–time, permitting multiple users to query a system adapted to their needs, using the query paradigm adapted to their problem such as query by example (QBE), browsing queries, or query by annotation.

User interaction is implemented over several levels and in diverse ways. Relevance feedback is implemented using positive and negative example images that can be used for a best–match QBE query. In contrast, browsing methods try to approach the searched image by giving overviews of the entire collection and by successive refinement. In addition to these query methods, Long term off line learning is implemented. It allows feature preferences per user, user domain or over all users to be learned automatically.

We present the *Viper* multimedia retrieval system as the core of the framework and an example of an MRML–compliant search engine. *Viper* uses techniques adapted from traditional information retrieval (IR) to retrieve multimedia documents, thus benefiting from the many years of IR research. As a result, textual and visual features are treated in the same way, facilitating true multimedia retrieval.

The MRML protocol also allows other applications to make use of the search engines. This can for example be used for the design of a benchmark test suite, querying several search engines in the same way and comparing the results. This is motivated by the fact that the content–based image retrieval community really lacks such a benchmark as it already exists in text retrieval.

1 Motivations

Content–based multimedia retrieval systems (CBMRSs) aim to retrieve documents from databases based on their similarities to a query composed of documents or document elements. The search paradigms are thus very different from that of conventional database queries, which rely on exact matches of specified fields and query terms.

There is as yet no standard query language for this paradigm, only the idea for a query taxonomy in content–based image retrieval (CBIR) has been presented in (Chang et al. 1999). SQL fulfills this role for conventional database queries. Such a language facilitates the system development, allowing components such as interfaces and benchmarking tools to be shared and reused. At present, every new CBMRS has its own interface, whether as a web version (IMedia Web Page 1999) or a locally installed version (QBIC Web Page 1998). Not only does this leads to duplication of effort, but it reduces system usability since users must learn the use of a new interface for each search engine. In §2, we propose MRML as a solution to this problem. The use of non–standardized interfaces also blocks the efforts to create a common benchmark in CBIR as it already exists in textual information retrieval (IR) (Vorhees & Harmann 1998). Measures for such a benchmark have already been developed (Müller et al 1999c)

Most existing CBMRSs treat only images, *e.g.* IBM's QBIC (Flickner et al. 1995), and define similarity using global and/or local color and texture features, whereas others use text as a starting point (Altavista Search Engine 1999). Some, *e.g.* Blobworld (Carson et al. 1997), allow the user to select and weight image segments when constructing a query. Much research has been devoted to developing image features for such queries, and to discovering means of combining and weighting them so as to produce results corresponding to the users' information needs. Very few such systems exploit the techniques and insights obtained during more than 40 years of IR research, despite the fact that they can often be applied or adapted to multimedia documents. *Viper*, a CBMRS inspired by IR, is presented in §3.

Little research has been done in the direction of combining all the elements of a CBMRS into a modular framework, where new modules can be added and changed based on the various fields of application. Such a distributed framework with its components is presented in §4.

2 Multimedia Retrieval Markup Language (MRML)

MRML is the result of a collaboration between the *Viper* group[1] at the University of Geneva and the CIRCUS group[2] at EPF Lausanne in cooperation with other research groups. The first aim of MRML was to separate the user interface from the query engine by specifying a structured and standardized way to send queries and to ease the cooperation between research groups.

2.1 Features of MRML

MRML is based on XML so that standard, freely–available parsers can be used. MRML is a multi paradigm protocol, offering features such as query by example (QBE), choice of databases, features or algorithms to use, and property sheets for specifying algorithm–specific parameters. It is extensible, so that private tags for special features of a system can easily be specified (even SQL can be embedded, if desired). Further details on MRML can be found in (Müller et al 1999).

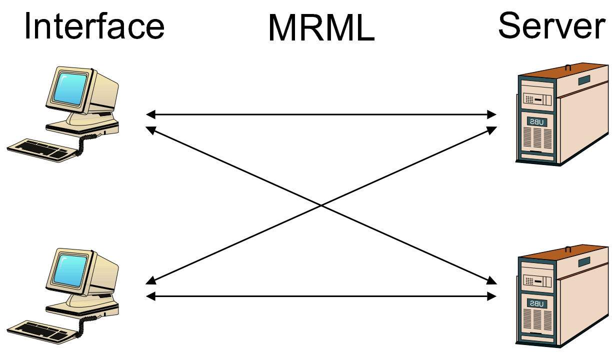

Figure 1: Communication with MRML

Such a protocol allows the reuse of user interfaces for new CBMRSs, and permits the user to connect to a variety of disparate query engines from the same interface, as illustrated in Figure 1. Not only does this reduce development time for CBMRSs and learning time for the user, but it also facilitates the comparison of query engines.

2.1.1 Description of MRML–based communication

MRML–based communications have the structure of a remote procedure call: the client connects to the server, sends a request, and stays connected to the server until the server breaks the

[1] Visual Information Processing for Enhanced Retrieval. `http://viper.unige.ch/`

[2] Content–based Image Retrieval and Consultation User–centered System. `http://lcavwww.epfl.ch/~zpecenov/CIRCUS/`

connection. The server shuts down the connection after sending the MRML message which answers the request.
MRML, in its current specification (and implementation) state, supports the following features:

- request of a capability description from the server,
- selection of a data collection classified by query paradigm; it is possible to request all collections which can be queried in a certain manner,
- selection and configuration of a query processor, also classified by query paradigm; MRML also permits the configuration of meta queries during run time,
- formulation of QBE queries,
- transmission of user interaction data.

2.1.2 Why XML for the description?

There are important reasons for using XML rather than a communication framework such as CORBA as a basis for the implementation of MRML. The first is that when using XML no large communication framework is necessary, as it is for CORBA. Secondly, MRML offers a common human–readable format for log files. Having a simple common format for user data will make it easier for research groups to share this type of data. Together with common free image collections (Annotated Groundtruth Database 1999), MRML–compliant systems will form a powerful tool for collecting and sharing CBIR user interaction data.
Another reason for the use of XML as a basis for MRML is the large number of free XML tools available such as parsers and tools to evaluate files in XML format. XML is about to become the main description language for all kinds of meta data on the internet and may also be used as content descriptors in MPEG–7 (MPEG Requirements Group 1998), thus ensuring the long term support of its specifications.

2.1.3 Graceful degradation

Graceful degradation is the key to successful independent extension of MRML. The basic principles can be summarized as follows:

- servers and clients which do not recognize an XML element or attribute encountered in an MRML text should completely ignore its contents,
- extensions should be designed so that all the standard information remains available to the generic MRML user.

These principles provide guidelines for independent extensions of MRML. To avoid conflicts between differing extensions of MRML, it is planned to maintain or promote a central database for the registration and documentation of MRML extensions. This would also facilitate the translation between user logs which contain extended MRML.

2.2 Examples for the Use of MRML

2.2.1 Logging onto a server

An MRML server listens on a port for MRML messages on a given TCP socket. When connecting, the client requests the basic properties of the server, and waits for an answer. The MRML code looks like this:

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

The server then informs the client of its capabilities.

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

Using similar simple messages, the client can request a list of 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.* meta query agents). The client can also request the algorithms which can be used with a given collection:

```
<mrml>
 <get-algorithms
  collection-id="collection-1" />
</mrml>
```

This request is answered by sending the corresponding list of algorithms. This handshaking mechanism allows both interactive clients and programs (such as meta query agents or automatic benchmarkers) to obtain information describing the server.
In a similar simple manner, the client can open and close sessions for a user, and configure the algorithms chosen by the user. This enables multi–user servers and also on–the–fly learning by the query processor.

2.2.2 Interface configuration

The client can then request property sheet descriptions from the server. Varying algorithms will have different relevant parameters which should be user–configurable (*e.g.* feature sets, speed vs. quality). *Viper*, for example, offers several weighting functions (Salton & Buckley 1987) and a variety of methods for, and levels of, pruning (Müller et al 1999a). All these parameters are irrelevant for other search engines. Thanks to MRML property sheets, the interface can adapt itself to these specific parameters. At the same time, MRML specifies the way the interface will turn these data into XML to send them back to the server.
Here is short example of an interface configuration:

```
<property-sheet
 property-sheet-id="sheet-1"
 type="numeric"
 numeric-from="1"
 numeric-to="100"
 numeric-step="1"
 caption="\% features evaluated"
 send-type="attribute"
 send-name="cui-percentage-features" />
```

This specifies a display element which will allow the user to enter an attribute with the caption "% of features evaluated". The values the user will be able to enter are integers between 1 and 100 inclusive. The value will be sent as an attribute *e.g.* `cui-percentage-features="33"`. This mechanism allows the use of complex property sheets, which can send XML text containing multiple elements.

2.2.3 Query formulation

The query step is dependent on the query paradigms offered by the interface and the search engine. MRML currently includes only QBE, but it has been designed to be extensible to other paradigms.
A basic QBE query consists of a list of images and the corresponding relevance levels assigned to them by the user. In the following example, the user has marked two images, the image `1.jpg` positive (`user-relevance="1"`) and the image `2.jpg` negative (`user-relevance="-1"`). All query images are referred to by their URLs.

```
<mrml session-id="1" transaction-id="44">
<query-step session-id="1"
 resultsize="30"
 algorithm-id="algorithm-default">
 <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 then return the retrieval result as a list of image URLs.
Queries can be grouped into transactions. This allows the formulation and logging of complex queries. This may be applied in systems which process a single query using a variety algorithms, such as the split–screen version of *TrackingViper* (Müller et al 1999b) or the system described by (Lee et al. 1999). It is important in these cases to preserve in the logs the knowledge that two queries are logically related one to another.

2.3 Further Usage of MRML

Although the initial idea for the use of the MRML protocol was to allow inter operability between CBMRSs, and thus the separation of interface and the actual query engine, it offers much more potential. Several applications can use the open interface to the server to get information and evaluate this information for example in form of a meta search engine or a benchmark.

2.3.1 Benchmarking CBIRS

As there is no common benchmark in the CBIR community yet as it already exists since more than ten years in the IR community (Vorhees & Harmann 1998), we are in the process of developing such a benchmark suite based on MRML. Thus a benchmark can be performed and evaluated automatically with little manual effort. Based on (Müller et al 1999c), we are developing a flexible benchmark test suite which takes relevance judgments into account and then compares the system responses of all available systems by several evaluation methods such as precision recall graphs, normalized averaged rank and precision after 20 images are retrieved (see Figure 2). The relevance judgments can either be clusters of images from manual classification or several judgments from a number of users. Experiments have shown that user judgments differ strongly (Squire et al 1999a, Squire & Pun 1997) which makes it important to allow several judgments to compare the system performance.
The time for the system response can also be a parameter for the evaluation; the usability of a system depends strongly on the response times, and systems which offer several methods to speed up the evaluation can be evaluated by these means.

2.3.2 Meta search engines

Meta search engines offer the possibility to automatically route one query to several search engines. One application is to help the user in finding the optimal search engine for the search problem, the other application is generating a combined solution from several query engines. A meta search engine for images, MetaSEEk, is described in (Beigi et al. 1998). A wide acceptance of a common query language like MRML would greatly simplify the development of such search engines.

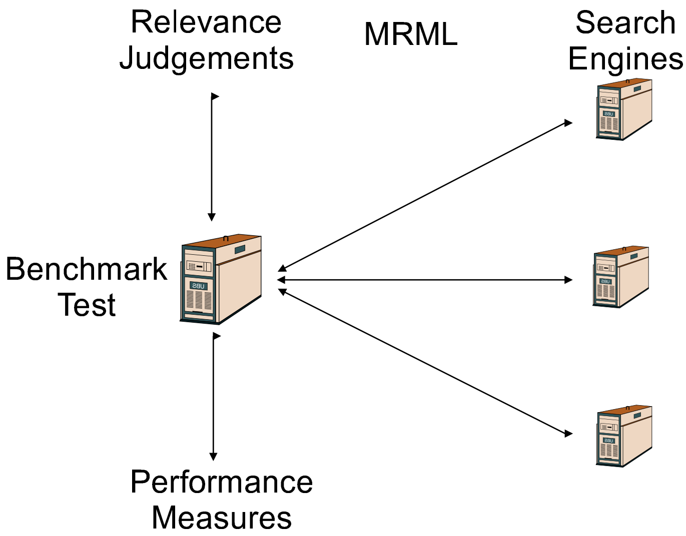

Figure 2: Benchmarking with MRML

3 *Viper* and SnakeCharmer

Viper and SnakeCharmer are two MRML–compliant components. Both are freely available to ease the use of MRML for research groups and the development of CBMRSs.

3.1 The Interface: SnakeCharmer

SnakeCharmer (Figure 3) is an MRML–compliant client application. It is written in JAVA for portability and offers query by multiple positive and negative examples, query history, multiple collection and algorithm selection, a scatter plot of the results according to various aspects of similarity and a basket for user–selected images.

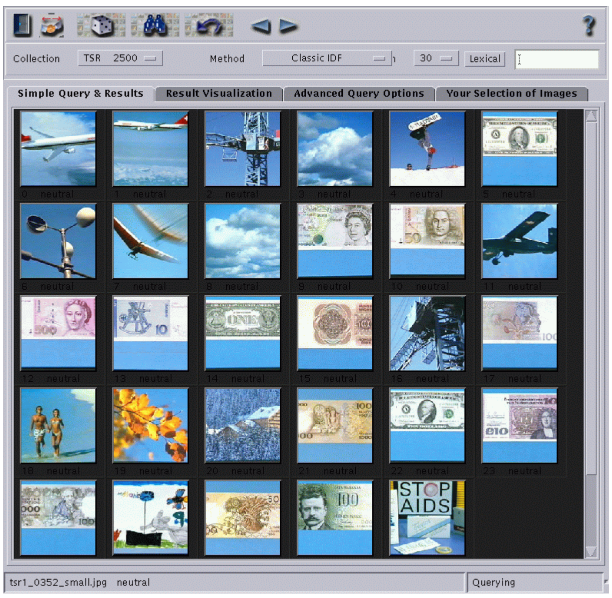

Figure 3: The JAVA interface SnakeCharmer

3.2 The *Viper* Search Engine

3.2.1 Search techniques

The *Viper* system (Müller et al 1999a, Squire et al 1999, Squire et al 1999a), inspired by IR systems, uses a very large number of simple features. In contrast to other CBIR systems *Viper* stores for each image a variable–length list of discrete features. A feature is considered as either present (with a certain frequency) or not present in the image. The resulting data structure can be treated like text: a text document is a variable–length sequence of words. Words are either present or not present in the text. *Viper* features are weighted similar to feature weighting schemes in text retrieval (Salton & Buckley 1988). One example for such a weighting scheme is given in Equations 1, 2.
At present more than 80,000 features are available to the system, of which each image has $\mathcal{O}(10^3)$ features. The mapping from features to images is stored in an inverted file. The use of such a data structure, in conjunction with a frequency–based feature weighting scheme makes the integration of text completely natural: textual features are treated in exactly the same way as visual ones. Thus multimedia documents have a natural representation in the system.
An example for such a weighting algorithm is given by

$$\text{feature relevance}_j = \frac{1}{N}\sum_{i=1}^{N}\left(tf_{ij}R_i\right) log^2\left(\frac{1}{cf_i}\right) \quad (1)$$

$$\text{image score}_{kq} = \sum_j \left(tf_{kj}\text{feature relevance}_j\right). \quad (2)$$

where tf is the term frequency of a feature in either a query or a result image, cf the collection frequency of a feature, q is a query with i=1..N input images, k a result image, j the number of a feature and R the relevance of a query image between [-1;1].
In principle, Equation 1 is based on the fact that a feature that is frequent in the collection does not distinguish images well from each other whereas a feature frequent in an image describes this image well.

3.2.2 Image features

The present version employs both local and global color and texture features extracted at several scales, and their frequency statistics in both the image and the whole collection. The intention is to make available to the system low–level features which approximate those present in the human vision system. More about the features can be read in (Squire et al 1999a).
The user can also add text to images and so facilitate the retrieval of these images using search by annotation.

3.2.3 Interactive feedback strategies

We have several choices for interactive feedback including relevance feedback as described in (Müller et al 2000a). We offer the user the possibility to mark several images as positive and negative examples. Since it is known that too much negative feedback causes problems in many CBIRSs, a mechanism to avoid such problems has been implemented, inspired by (Rocchio 1971).
In contrast to classical relevance feedback, image browsing attacks the problem of finding a good example image as starting point for a QBE search. An image browser helps the user in finding a target by presenting him a sequence of overviews of the database which are successively refined by learning from user interaction. The overviews presented to the user are chosen so as to maximize the expected information gain. This technique was first employed and evaluated by `PicHunter` (Cox et al. 1996). In (Müller et al 1999b) we presented *TrackingViper*, an extension and modification of the `PicHunter` scheme to give the user the possibility to change his mind during the query process. *TrackingViper* is incorporated into the *Viper* framework.

3.2.4 Long term learning

The use of usage log files of the system permits long term learning as described in (Müller et al 2000). This learning technique only takes into account images which are marked together in the same query step. Images marked together positively associate their common features with a higher weighting. The features common in images marked together positive and negative in the same query step by contrary get a lower weighting. This learning technique can be employed on a user, a domain and an overall basis to take into account various goals of differing users or user groups.

4 The Open Multimedia Framework

This section describes the framework and the exchangeable parts of our system

4.1 The Framework

Rather than having a closed system for the retrieval of multimedia documents we would like to have an open platform where components can be exchanged easily and fast without having to change the system. This helps to integrate several research groups of neighboring disciplines. Research groups can thus work on the same project and just do their parts with defined programming interfaces.

For making this framework possible, MRML has been developed and used in cooperation between the *Viper* group, the CIRCUS group and in collaboration with the MIRROR group[3] at the University of Twente. The separation of interface and search engine is already achieved with *Viper* and CIRCUS on the basis of MRML, other parts of the system can also be replaced and expanded easily without changes in the main system. In Figure 4, we see a graphical representation of our CBMR framework. Within this framework, it is easy to adapt the system to divers requirements and use it in various domains (medical, trademarks, consumer photographs). The replaceable components of our project like the various algorithms for querying, multimedia collections and characteristic features are described in the following sections.

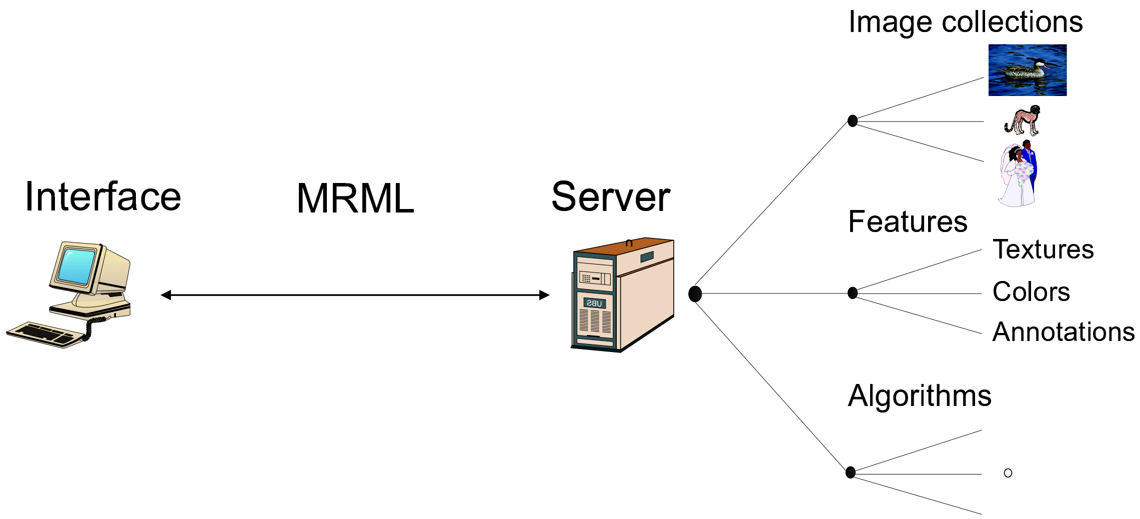

Figure 4: The complete Framework of the *Viper* system

The results of this research will be made freely available. The JAVA–based interface Snake-Charmer in conjunction with an MRML parser and the CBMRS *Viper* will be placed in the

[3]Multimedia Information Retrieval Reducing information OveRload.
`http://wwwhome.cs.utwente.nl/~arjen/mmdb.html`

public domain to allow researchers to concentrate on those aspects of the problem which interest them most.
Work is continuing on the development of a freely available database (Annotated Groundtruth Database 1999) and query set for an MRML–based benchmarking suite. The distribution of MRML in conjunction with such a benchmark could make possible a first comparison of a large number of CBMRSs.

4.2 Exchanging Collections

The *Viper* package includes a simple command to add a new image collection to its query engine. The chosen features are extracted automatically and an inverted file is generated with these features. The result is then added to the collections of a query engine. In the interface all possible image databases are displayed and the wanted collection can be chosen. At the moment we have about 15 different collections of images incorporated into the system. Our system also offers the possibility to include collections of multimedia items such as web pages, thus extending the concept of image search.

4.3 Exchanging Features

We have different sets of features for the *Viper* system. Each combination of features can be selected for a query. It is as well possible to have various feature sets for different collections, which is absolutely necessary when working in several domains or with multi modal data. Medical images need for example different feature sets than face image or photograph collections. Thus with each image collection the interface is configured to display the possible feature groups. At the moment we offer besides text local and global color features and local and global texture features based on Gabor Wavelets. The investigation of feature groups for special images like trademarks is planned.

4.4 Exchanging Algorithms

At the moment, we have several weighting schemes incorporated into the system. These weighting schemes are described in (Salton & Buckley 1988) and a comparison of the weighting schemes for CBIR can be found in (Squire et al 1999).
As different domains might need varying weightings and varying access to the features, this functionality is offered. Examples for weighting schemes we use are:

- classical inverse document frequency (see Equations 1 and 2),
- best weighted probabilistic,
- binary term independence.

4.5 Distribution of the Computation

The use of Corba allows us to distribute the actual computation on a number of computers in a network. For the same query image, various collections can be queried on several machines, or various feature groups can be queried on several machines depending on the computing power available (see Figure 5).
This allows to balance the computational load which is important when a system is used intensively and by several users.

5 Conclusion

This paper describes a new, open framework for multimedia retrieval which offers easy adaptability of the system to divers areas of application. Within this framework, developers of retrieval

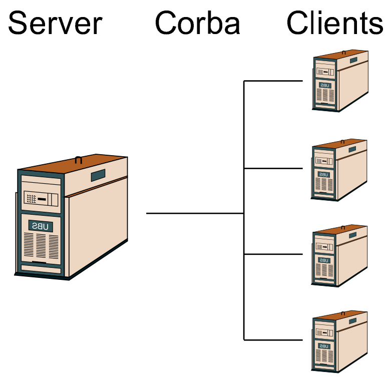

Figure 5: Distribution of the computing with Corba.

systems will be able to freely use and exchange proven interfaces and the user can chose and compare several retrieval engines. The use of a common communication protocol by many CBMRSs will greatly facilitate the comparison of systems, and, associated with a common database, is a first step towards a CBMR benchmarking suite.
The multimedia retrieval engine *Viper* has demonstrated the feasibility of MRML, and various techniques adapted from IR have been shown to be effective in CBMR. Textual and visual features have a natural common representation in *Viper*, leading to their easy integration for true multimedia retrieval.
The *Viper* framework allows for an easy expansion and exchange of every part of the system. Thus it is possible to adapt the system to almost every scenario where multimedia data needs to be managed. Adaptations to various domains like medical imaging and trademark recognition are underway and need to be evaluated.
With the distribution of the system via Corba, we complement the separation of the interface and the search engine via MRML by the distribution of the actual computation over several machines on the internet or a local network.

6 Acknowledgments

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

References

Altavista Search Engine (1999). Altavista image search engine: `http://www.altavista.com/`.

Annotated Groundtruth Database (1999). Department of Computer Science and Engineering, University of Washington `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`.

Beigi, M., Benitez, A. B. & Chang, S.-F. (1998). Metaseek: A content-based meta-search engine for images, *Symposium on Electronic Imaging: Multimedia Processing and Applications - Storage and Retrieval for Image and Video Databases VI, IST/SPIE'98, San Jose, CA*.

Carson, C., Belongie, S., Greenspan, H. & Malik, J. (1997). Region-based image querying, *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, IEEE Computer Society, San Juan, Puerto Rico.

Chang, Y.-C., Bergmann, L., Smith, J. R. & Li, C.-S. (1999). Query taxonomy of multimedia databases, *in* Panchanathan et al. (1999). (SPIE Symposium on Voice, Video and Data Communications).

Cox, I. J., Miller, M. L., Omohundro, S. M. & Yianilos, P. N. (1996). Target testing and the PicHunter Bayesian multimedia retrieval system, *Advances in Digital Libraries (ADL'96)*, Library of Congress, Washington, D. C., pp. 66–75.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. & Yanker, P. (1995). Query by image and video content: The QBIC system, *IEEE Computer* **28**(9): 23–32.

Imedia Web Page (1999). Surfimage webdemo http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi.

Lee, C. S., Ma, W.-Y. & Zhang, H. (1999). Information Embedding Based on User's Relevance Feedback for Image Retrieval, *in* Panchanathan et al. (1999). (SPIE Symposium on Voice, Video and Data Communications).

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

Müller, H., Müller, W., Marchand-Maillet, S., Squire, D. M. & Pun, T. (2000). Long term learning in content-based image retrieval, *Technical Report 00.04*, Computer Vision Group, Computing Centre, University of Geneva, rue Général Dufour, 24, CH-1211 Genève, Switzerland.

Müller, H., Müller, W., Squire, D. M., Marchand-Maillet, S. & Pun, T. (2000a). Strategies for positive and negative relevance feedback in image retrieval, *Technical Report 00.01*, Computer Vision Group, Computing Centre, University of Geneva, rue Gnral Dufour, 24, CH-1211 Genve, Switzerland.

Müller, H., Müller, W., Squire, D. M. & Pun, T. (1999c). Performance evaluation in content-based image retrieval: Overview and proposals, *Technical Report 99.05*, Computer Vision Group, Computing Centre, University of Geneva, rue Gnral Dufour, 24, CH-1211 Genve, Switzerland.

Müller, H., Squire, D. M., Müller, W. & Pun, T. (1999a). Efficient access methods for content-based image retrieval with inverted files, *in* Panchanathan et al. (1999). (SPIE Symposium on Voice, Video and Data Communications).

Müller, W., Pečenović, Z., de Vries, A. P., Squire, D. M., Müller, H. & Pun, T. (1999). MRML: Towards an extensible standard for multimedia querying and benchmarking – Draft proposal, *Technical Report 99.04*, Computer Vision Group, Computing Centre, University of Geneva, rue Général Dufour, 24, CH-1211 Genève, Switzerland.

Müller, W., Squire, D. M., Müller, H. & Pun, T. (1999b). Hunting moving targets: an extension to Bayesian methods in multimedia databases, *in* Panchanathan et al. (1999). (SPIE Symposium on Voice, Video and Data Communications).

Panchanathan, S., Chang, S.-F. & Kuo, C.-C. J. (eds) (1999). *Multimedia Storage and Archiving Systems IV (VV02)*, Vol. 3846 of *SPIE Proceedings*, Boston, Massachusetts, USA. (SPIE Symposium on Voice, Video and Data Communications).

QBIC (1998). QBIC™ – IBM's Query By Image Content http://wwwqbic.almaden.ibm.com/~qbic/.

Rocchio, J. J. (1971). Relevance feedback in information retrieval, *The SMART Retrieval System, Experiments in Automatic Document Processing*, Prentice Hall, Englewood Cliffs, New Jersey, USA, pp. 313–323.

Salton, G. & Buckley, C. (1987). Term weighting approaches in automatic text retrieval, *Technical Report 87-881*, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501.

Salton, G. & Buckley, C. (1988). Term weighting approaches in automatic text retrieval, *Information Processing and Management* **24**(5): 513–523.

Squire, D. M., Müller, W. & Müller, H. (1999). Relevance feedback and term weighting schemes for content-based image retrieval, *in* D. P. Huijsmans & A. W. M. Smeulders (eds), *Third International Conference On Visual Information Systems (VISUAL'99)*, number 1614 in *Lecture Notes in Computer Science*, Springer-Verlag, Amsterdam, The Netherlands, pp. 549–556.

Squire, D. M., Müller, W., Müller, H. & Raki, J. (1999a). Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback, *The 11th Scandinavian Conference on Image Analysis (SCIA'99)*, Kangerlussuaq, Greenland, pp. 143–149.

Squire, D. M. & Pun, T. (1997). A comparison of human and machine assessments of image similarity for the organization of image databases, *in* M. Frydrych, J. Parkkinen & A. Visa (eds), *The 10th Scandinavian Conference on Image Analysis (SCIA'97)*, Pattern Recognition Society of Finland, Lappeenranta, Finland, pp. 51–58.

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