MRML: An Extensible Communication Protocol for Interoperability and Benchmarking of Multimedia Information Retrieval Systems

Wolfgang Müller*, Henning Müller*, Stéphane Marchand-Maillet*, Thierry Pun*,
David McG. Squire[1], Zoran Pečenović[2], Christoph Giess[3], Arjen P. de Vries[4]

*Computer Vision Group, Computer Science Department, University of Geneva, Switzerland.
[1]Computer Science and Software Engineering, Monash University, Melbourne, Australia
[2]LCAV and Ergonomics Group, Ecole Polytechnique Fédérale de Lausanne, Switzerland
[3] Medical and Biological Informatics, Deutsches Krebsforschungszentrum, Heidelberg, Germany
[4]CWI, Amsterdam, The Netherlands

ABSTRACT

While in the area of relational databases interoperability is ensured by common communication protocols (*e.g.* ODBC/JDBC using SQL), Content Based Image Retrieval Systems (CBIRS) and other multimedia retrieval systems are lacking both a common query language and a common communication protocol.

Besides its obvious short term convenience, interoperability of systems is crucial for the exchange and analysis of user data. In this paper, we present and describe an extensible XML-based query markup language, called MRML (Multimedia Retrieval Markup Language). MRML is primarily designed so as to ensure interoperability between different content–based multimedia retrieval systems. Further, MRML allows researchers to preserve their freedom in extending their system as needed.

MRML encapsulates multimedia queries in a way that enables multimedia (MM) query languages, MM content descriptions, MM query engines, and MM user interfaces to grow independently from each other, reaching a maximum of interoperability while ensuring a maximum of freedom for the developer. For benefiting from this, only a few simple design principles have to be respected when extending MRML for one's private needs. The design of extensions within the MRML framework will be described in detail in the paper.

MRML has been implemented and tested for the CBIRS Viper, using the user interface SnakeCharmer. Both are part of the GNU project and can be downloaded at our site*.

Keywords: CBIRS, MRML, interoperability, communication protocol, evaluation

1. INTRODUCTION

During the past decade interest and research in CBIR has flourished. One of the attractions of CBIR research is that it is multidisciplinary. A wide range of research issues must be addressed: human perception, human-computer interaction, interface design, query representation for content-based search, feature extraction and representation, indexing structures, term-weighting schemes, pruning strategies, performance evaluation and more. This richness is also one of the great challenges for CBIR research. At present, a researcher new to the field must build an entire system before he or she can address the aspect of the problem which interests him or her most. A researcher working alone or in a small team will have great difficulty constructing a state of the art system during the typical time-frame of a PhD or post-doc. MRML, the Multimedia Retrieval Markup Language, provides a framework in which components of CBIR systems can be built and exchanged interoperably. If state of the art components of CBIR systems are made MRML-compliant, all researchers will benefit from the opportunity to focus their investigations on their area of greatest interest and expertise.

Presently, almost every content-based image retrieval system (CBIRS) is a hard-wired connection between an interface and the functional parts of a program. Some programs provide easy-to-use web interfaces,[1] while others

mailto:Wolfgang.Mueller@cui.unige.ch
*http://www.mrml.net/download

need to be installed locally[2] and may be specific to particular operating systems. The reuse of components in CBIR, *e.g.* of user interfaces, is thus very sparse. This is not only a time-consuming problem, since everything needs to be developed anew for each system, but it makes the sharing of user data and the comparison of system performances difficult.

In order to address these problems, Y.-C. Chang *et al.*[3] proposed a query taxonomy for multimedia databases. They proposed an initial formulation of the requirements for a system enabling communication between multimedia databases and clients. However, this approach is not yet translated into an extensible protocol.

In this paper we present MRML, an XML-based markup language for multimedia queries. MRML was designed to facilitate a bottom-up development approach, which separates the communication problem from the search for the best query language (*e.g.*[4,5]) for multimedia databases. In other words, not only it is designed to fulfill the short-term needs of the image database research community, but it is also designed to cater for its long-term needs.

The development of standard query languages, together with standard methods for transmitting queries and data, can improve the interoperability of CBIRSs and thus increase the use and usefulness of multimedia databases. SQL and ODBC are examples of such developments for relational databases. The aim of MRML, however, is more similar to that of the DICOM protocol,[6] which promoted the interoperability of medical imaging systems from different vendors. We address the urgent need for common tools which will facilitate the development and evaluation of multimedia database systems. By this means, we facilitate the development of common benchmarks for CBIRS performance, similar those used for textual information retrieval.[7]

The query-by-example (QBE) paradigm with relevance feedback (including browsing) is the search paradigm employed by most current CBIRSs. We therefore provide an extensible QBE facility within MRML. Further, some MRML-compliant tools have been developed and made freely available* under the GNU Public License. They have now been accepted as official part of the GNU project†. These are described briefly in Section 2, and include a CBIR search engine (*Viper*), which acts as a server, and an interface (SnakeCharmer), which acts as a client. Scripts (mostly Perl scripts) have also been made available, which might provide a basis for the creation of standard CBIRS benchmarks. An overview of various evaluation methods is given in,[8] where the use of freely-available annotated image collections, such as,[9] as test datasets is also advocated.

In order to be useful for research, MRML needs to be a "living standard": research groups will need to be able to test and use extensions without having to ask a committee for approval. We therefore employ a development model which permits phases of independent growth with subsequent code merging. In section 3, we present the main features of MRML and, in section 4, we show an example of how MRML can be extended to suit particular needs while staying coherent with the common standard.

Currently we are using MRML as the communication protocol for a flexible benchmarking system supporting multiple types of interaction, as described in the last section and in another article within this volume.[10]

2. *Viper*, CIRCUS AND SnakeCharmer

MRML was initially designed to facilitate cooperation between research groups. The main programs for our testbed originate from the Ecole Fédérale Polytechnique de Lausanne (CIRCUS and SnakeCharmer) and from the University of Geneva (*Viper*). In this testbed, we use MRML to link a single interface (SnakeCharmer) to two different CBIRS (CIRCUS and *Viper*). A link to the MIRROR DBMS[11] is currently under development.

Viper‡ is an image search engine based on techniques commonly used in text retrieval and thus offers efficient access to a very large number of *possible* features (more than 80,000 simple color and texture features, both local and global). Detailed descriptions of *Viper* may be found in.[12,13]

CIRCUS§ is a server framework supporting multiple image retrieval methods and algorithms. Currently 4 methods based on LSI[14] and wavelet decomposition are supported.

SnakeCharmer 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.

†`www.gnu.org`

‡`http://viper.unige.ch/`

§`http://lcavwww.epfl.ch/CIRCUS`

3. MULTIMEDIA RETRIEVAL MARKUP LANGUAGE

MRML[¶] is formally specified in.[15] It provides a framework that separates the query formulation from the actual query shipping. It is designed to markup multi-paradigm queries for multimedia databases. MRML enables the separation of interface and query engine and thus eases their independent development.

MRML can be embedded into an existing system with little effort. First, it is XML-based, meaning that standard parsers can be used to process the communication messages. Further, the code for an example MRML-compliant CBIR system is freely-available and provides the basic implementation of both ends of an MRML-based communication toolkit. MRML is currently in a testing phase at several universities and further applications based on this protocol such as benchmark systems and meta-query engines are under development.

MRML is designed to allow extension by independent groups. By this means, it provides a research platform for extensions which later may become a part of common MRML.

3.1. Design goals of MRML

It is important for the following sections to keep in mind the priorities which we took into account during the design of MRML.

Interoperability: Interoperability is an obvious short term need of the CBIRS community. The fact that the interface between CBIRS client and server is not specified hampers research. Topics that could benefit from interoperability include:

- Meta-query engines query several "normal" query engines and assemble the results.[16] Constructing a meta-query engine would require to define a protocol abstraction layer corresponding to each of the different query embedded in the system. Using a common protocol would save a substantial amount of work.
- Human-computer-interaction aims at comparing the impact of different user interfaces on the performance of identical query engines, or test several engines with the identical interfaces. In this context, by ensuring the compatibility between engines and interfaces, MRML would ease this type of evaluation.
- Evaluation of query engines: Thanks to MRML, one can design a benchmark package that connects to a server, sends a set of queries and evaluates the results.

Extensibility without administration overhead: it was our goal to provide a communication protocol which can be extended without having to ask a standardization body for permission. MRML enables independent development of extensions. As we will describe in Section 4.2 we invite MRML users to render their extensions accessible at `http://www.mrml.net/extensions/`. Later, stable extensions can be added to new common versions of MRML.

Common log file format: The whole area of CBIRS is craving for ground truth or other user data. MRML provides a common, human readable, easy to analyze, format for logging communication between CBIRS client and server. MRML contains a maximum of data which might be of interest for computer learning purposes. If needed, extensions of MRML can be designed in order to send additional data.

Simplicity of implementation: everything was designed so as to minimize the implementation overhead incurred when using MRML, while keeping a maximum of flexibility. MRML is well–formed DTD-less XML. This eliminates the need for validating parsers while maintaining all the flexibility that is needed.

3.2. Features of MRML

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 connection. The server shuts down the connection after sending the MRML message which answers the request. This connectionless protocol has the advantage of easing the implementation of the server. To limit the performance loss caused by frequently reconnecting, it is possible to send several requests as part of a single MRML message. The extension of MRML to a protocol permitting the negotiation of a permanent connection is also planned.

MRML, in its current specification (and implementation) state, supports the following features:

[¶] `http://www.mrml.net`

- request of a capability description from the server,
- opening and closing sessions with the server,
- selection of a data collection classified by query paradigm; it is possible to request 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 run time configuration of meta-queries,
- formulation of QBE queries,
- transmission of user interaction data.

The final feature reflects our strong belief that affective computing[17] will soon play a role in the field of content-based multimedia retrieval. MRML already supports this by allowing the logging of user interaction data.

Graceful degradation: independent development on a common base 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 (see examples in Section 4).

These principles provide guidelines for independent extensions of MRML.

To avoid conflicts between differing extensions of MRML, and in order to we plan 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.

3.3. Logging onto a CBIR sever

Logging on a CBIR server, as well as the configuration has been described in the specifications.[15]

3.4. 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 images, again represented by their 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*[18] or the system described by Lee *et al.*.[19] It is important in these cases to preserve in the logs the knowledge that two queries are logically related one to another.

4. EXTENDING MRML

In this section, we demonstrate the flexibility of MRML. After a small example that illustrates the design guidelines which should be followed in designing extensions, we show how MRML can be extended towards region queries. Then we show how easily MRML and MPEG-7 multimedia descriptions can be interleaved.

4.1. A region query extension for MRML

In the past, most CBIR systems concentrated on queries for complete images. One of the first systems to use regions was *Blobworld*.[20] Here, the user is presented with a segmented image. He/she can then specify which of the segments (blobs) is relevant to the query.

Here we give an example how this can be expressed in MRML, followed by an explanation:

```
<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/banknote.jpg"
                                    user-relevance="1">
                <!-- interested in foreground -->
                <cui-relevance-region cui-region-start-x="21"
                                      cui-region-start-y="12"
                                      cui-region-outline="e216s137w216n137"
                                      cui-user-relevance="1"/>
                <!-- but not interested in background -->
                <cui-relevance-region cui-region-start-x="21"
                                      cui-region-start-y="12"
                                      cui-region-outline="s137e216n137w216"
                                      user-relevance="-1"/>
            </user-relevance-element>
            <user-relevance-element image-location="http://viper.unige.ch/2.jpg"
                                    cui-user-relevance="-1">
                <cui-relevance-region region-id="ad8b" user-relevance="-1"/>
            </user-relevance-element>
        </user-relevance-list>
    </query-step>
</mrml>
```

The example contains a QBE query augmented by segment information. The `user-relevance-element`s contain the usual information contained in a `user-relevance-element` plus, `cui-relevance-region`s. The benefit of this, is that clients that are not region–aware still receive and understand the main parts query (graceful degradation).

A `cui-relevance-region` contains either the `region-id` of a previously specified region, or a chain code[21] giving the outline of the region. Chain codes specify regions by giving a starting point and giving directions from this starting point. In the case of the first `cui-relevance-region` this is: 137 pixels south, 216 east, 137 north and 216 west, marking a rectangular region. Defining the region to the right of the moving direction as inside, we can also describe inverse segments as it is done in the second `cui-relevance-region`. Fig. 1 describes the banknote contained in the image as relevant, but the background as not relevant: requested are banknotes with a different background.

4.1.1. Classical query languages and MRML

Simply use a CDATA section and express the query using the query language chosen:

```
<mrml session-id="1" transaction-id="44">
    <query-step session-id="1"
                resultsize="30"
                algorithm-id="algorithm-default">
```

Figure 1. *Example of an MRML extension. The user has requested a banknote (marked white) with a background different from the one presented (marked black).*

```
        <cui-prolog-query result-container="ResultList" cui-relative-weight="0.1">
           <![CDATA[ findall(Image,contains(banknote,Image),ResultList). ]]>
        <cui-prolog-query/>
        <user-relevance-list cui-relative-weight="0.9">
            <user-relevance-element image-location="http://viper.unige.ch/banknote.jpg"
                                    user-relevance="1"/>
        </user-relevance-list>
    </query-step>
</mrml>
```

This example performs a QBE query using the banknote image shown in Fig. 1, but without regions. It also queries a base of prolog facts for images which contain the Prolog atom banknote. The query results will be mixed, weighting the rank in the Prolog query with 0.1, and the rank obtained in the QBE query with 0.9. Of course, designers who want to share this use of MRML have to document the internals.

4.1.2. MRML and MPEG-7

MPEG-7, the multimedia content description interface, is a large collection of description schemes, aiming at describing all aspects of multimedia data. Currently, MPEG-7 data descriptions are given in XML Schema. The XML Schema is considered as the replacement of the XML DTD (Document Type Definition), *i.e.* it specifies a grammar for the text. While the DTD is geared towards text markup, XML Schema is geared towards data transfer and allows the definition of type systems.

We did not choose XML–Schema as a basis for MRML as the specification of XML–Schema is not yet stable, and —at the time of writing— there are no parsers that meet every aspect of the XML–Schema specification. However, any text which contains XML-Schema is also well–formed XML. As a consequence, there is no problem in piping MPEG-7 multimedia content descriptions through MRML, for example in a query:

```
<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/banknote.jpg"
                                    user-relevance="1"/>
            <mpeg-7>
                <!-- put some MPEG-7 text here -->
            </mpeg-7>
        </user-relevance-list>
    </query-step>
</mrml>
```

Again, a query processor able to process MPEG-7 will recognize the `mpeg-7` tag and act accordingly. When not recognized, the tag is simply ignored. Using this method the structured annotation DS was demonstrated on the MPEG-7 Geneva meeting.

4.1.3. MRML and binary data

MRML's preferred mechanism for transferring binary data is to send the URL where the data can be found. Binary data is then retrieved using the URL. As it is a primary goal of MRML to enable the sharing of logging data we suggest to transfer big chunks of data as follows.

Binary data which stays constant over several sessions (*i.e.* images and other media items contained in the queried collection) should be transfered using their URL, as described above. This keeps log files relatively small, yet data is accessible for everyone.

Binary data which changes during the query process (*e.g.* a file containing an example image for a QBE query which is not accessible by the web) should be transferred using two attributes. One of the attributes should contain the base64-encoded binary data, the other one the corresponding MIME type.

However, in most cases, it is preferable to design proper extensions to MRML which provide the best accessibility and readability of the resulting logs.

4.2. The MRML development model

As it has been stated throughout this article, MRML allows each developer to extend MRML according to his/her needs. In particular, these extensions can coexist, and a notification of a central body is not necessary for making these extensions work. However, to maximize the usefulness of MRML we are presently setting up a database which contains documentation of extensions to MRML. It is also intended to provide a forum for groups which want to extend MRML into similar directions.

We propose to develop extensions to MRML in the following fashion. Search first the page `http://www.mrml.net/extensions/` for documentation of extensions which might already do what you want.

- If so,
  1. implement the existing extension;
  2. double-check with the author of the existing extension that the documentation has been understood in the right way;
  3. add your name and your affiliation to the list of people/groups who are using this extension which is kept on `www.mrml.net`.
- If not,
  1. implement the extension;
  2. submit documentation for your extension along with your name and your affiliation to `www.mrml.net`.

The information contained on `http://www.mrml.net/extensions/` will be useful both for analyzing logs and for *merging* extensions, once an extension has proven more useful than others.

5. FURTHER USE OF MRML

In this article, we have presented a stable, extensible and useful framework for the use in CBIRS and other multimedia retrieval systems. The following two sections describe possible uses of MRML.

5.1. Benchmarks

Only preliminary steps have been taken by the CBIR community towards developing common benchmarks – a comparison of evaluation techniques may be found in.[22] In Fig. 2 we give an UML sequence diagram which describes our benchmarking system.

MRML serves here for the separation between benchmarking system (BS) and benchmarked system (S). At the beginning of the benchmark the BS will establish a connection to the S using MRML. It then configures a session with a given algorithm (this is an input parameter of the BS). The Relevance Data Repository contains the results of queries which have been evaluated by hand by a number of test users. The BS now sends for each test user and each test query an MRML query to the S. It then uses the relevance data for generating a feedback query, which again are posed to the S. Using these data we can evaluate the learning capacities of S. All query results are stored in a result repository, allowing comparison of different runs and different query engines.

Replacing the QBEInitalQueryFormulator and QBEFeedbackGenerator by other components permits implementing other feedback strategies. While in benchmarking of QBE systems the images are considered either as being relevant or nonrelevant, in the cases of image browsers this notion is not sufficient. We therefore to support the notion of relative relevance. How one can extend the benchmarking system to support this notion of relevance is described elsewhere in this volume.[10]

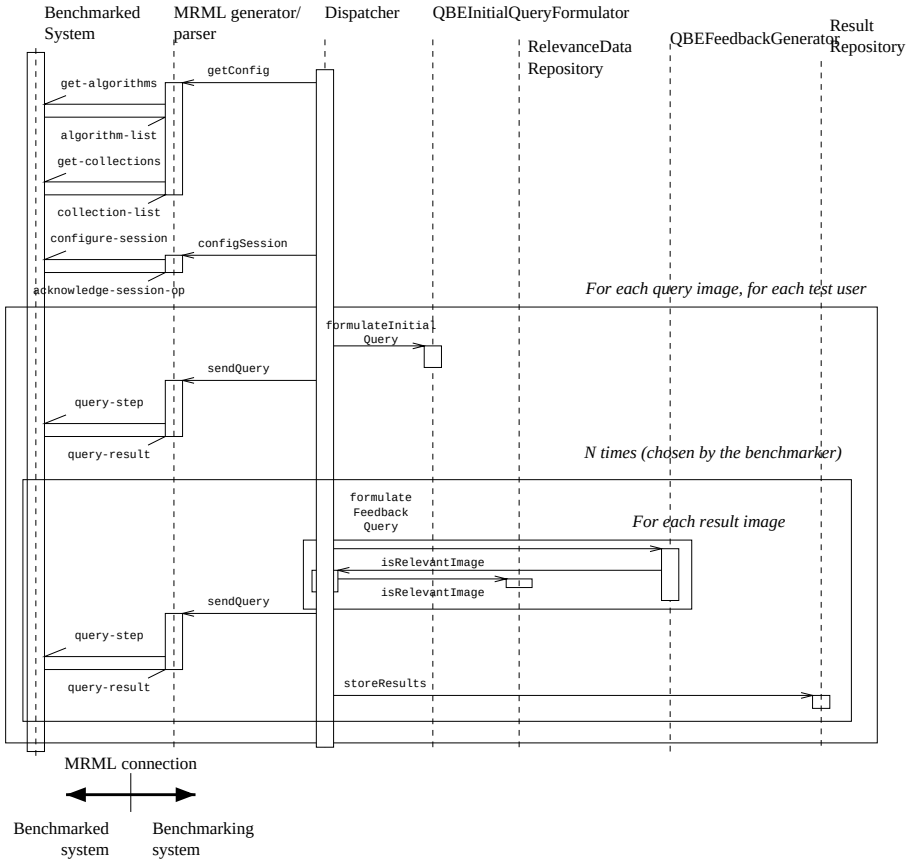

Figure 2. *Sequence diagram for our benchmarking architecture. Here, MRML is used for the communication between benchmarking system and benchmarked system. All components, except for the component marked Benchmarked System are part of the benchmarking system.*

5.2. Meta query engines

We are currently conceiving a meta query engine which queries MRML compliant servers. We plan to use methods similar to the ones described in.[16]

In the following, we will describe a meta query engine for QBE queries. We will call the Meta Query Engine *MQE*. The *MQE* will query a set of other query engines, QE_1 through QE_n.

Initialization: when started, the *MQE* will query QE_1 through QE_n for the list of available algorithms (query methods) and collections (using the MRML signals `get-algorithms`, `get-collections`).

Opening a session: on connection with the client, MQE will synthesize a list of collections and algorithms available on the different *QE*. Here we have the choice between different levels of elaboration:

- The most simple *MQE* will just give the user the choice between the different *QE*, handing through their choices of algorithms and collections, as well the property sheets for the configuration of algorithms.
- A more complex *MQE* will build a complex property sheet, enabling the distribution of one query on multiple query processors at once, giving full configuration choice to the user.

The *MQE* will not yet open a session on the $QE_{1\ldots n}$.

Before the first query of the *MQE*, the *MQE* will open sessions on each of the $QE_{1\ldots n}$ that has been configured by the user.

On each query the *MQE* will now pass through the queries (`query-step`) to each *QE* configured by the user. It will receive a *query-result* as an answer by each query engine and merge these results, passing the merged results to the client.

6. CONCLUSION

The development of MRML and the first MRML-compliant tools have established a common framework for the fast development of CBIR applications. To our knowledge, MRML is the first general communication protocol for CBIR actually implemented. The source code for the interface and the query engine is freely available under GPL*. This helps developers of retrieval engines and developers of user interfaces to develop complete systems on the basis of existing components. Extensive tests have shown the stability of the protocol and our test components.

Since MRML is a free and extensible standard, the availability of more applications and tools supporting such a protocol will further facilitate the development of CBIR applications supporting a diversity of query paradigms.

More important in our opinion is the fact that the adoption of MRML will lead to the possibility of comparing different CBIR applications objectively. We have developed a system that will make it easy to benchmark all MRML-compliant systems in a way similar the benchmark which exist in the database and information retrieval communities. Finally, the possibility of sharing MRML user logs will provide a useful tool for the sharing of user interaction data for learning and evaluation purposes.

Acknowledgments

This project is supported by the Swiss National Foundation for Scientific Research under grant number 2000-052426.97.

REFERENCES

1. "Surfimage webdemo." `http://www-rocq.inria.fr/cgi-bin/imedia/surfimage.cgi`, 1999.
2. "QBIC^{TM} – IBM's Query By Image Content." `http://wwwqbic.almaden.ibm.com/~qbic/`, 1998.
3. Y.-C. Chang, L. Bergmann, J. R. Smith, and C.-S. Li, "Query taxonomy of multimedia databases," in Panchanathan *et al.*.[23] (SPIE Symposium on Voice, Video and Data Communications).
4. J. Z. Li, M. Özsu, D. Szafron, and V. Oria, "MOQL: A Multimedia Object Query Language," in *The Third International Workshop on Multimedia Information Systems*, pp. 19–28, (Como, Italy), September 1997.
5. A. de Vries, "Mirror: Multimedia query processing in extensible databases," in *Proceedings of the fourteenth Twente workshop on language technology (TWLT14): Language Technology in Multimedia Information Retrieval*, pp. 37–48, (Enschede, The Netherlands), December 1998.
6. B. Revet, *DICOM Cook Book for Implementations in Madalities*, Philips Medical Systems, Eindhoven, Netherlands, 1997.

7. 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.
8. 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* , 2000.
9. "Annotated groundtruth database." Department of Computer Science and Engineering, University of Washington, `http://www.cs.washington.edu/research/imagedatabase/groundtruth/`, 1999.
10. W. Müller, S. Marchand-Maillet, H. Müller, and T. Pun, "Towards a fair benchmark for image browsers," in *Internet Multimedia Management Systems*, N.N., ed., vol. 4210 of *SPIE Proceedings*, (Boston, Massachusetts, USA), November 6–8 2000. (SPIE Information Technologies 2000).
11. A. de Vries, M. van Doorn, H. Blanken, and P. Apers, "The Mirror MMDBMS architecture," in *Proceedings of 25th International Conference on Very Large Databases (VLDB '99)*, pp. 758–761, (Edinburgh, Scotland, UK), September 1999. Technical demo.
12. H. Müller, D. M. Squire, W. Müller, and T. Pun, "Efficient access methods for content-based image retrieval with inverted files," in Panchanathan *et al.*.[23] (SPIE Symposium on Voice, Video and Data Communications).
13. D. M. Squire, W. Müller, H. Müller, and J. Raki, "Content-based query of image databases, inspirations from text retrieval: inverted files, frequency-based weights and relevance feedback," in *The 11th Scandinavian Conference on Image Analysis (SCIA'99)*, pp. 143–149, (Kangerlussuaq, Greenland), June 7–11 1999.
14. Z. Pečenović, "Image retrieval using latent semantic indexing," final year graduate thesis, AudioVisual Communications Lab, Ecole Polytechnique Fédérale de Lausanne, Switzerland, June 1997.
15. W. Müller, Z. Pečenović, A. P. de Vries, D. M. Squire, H. Müller, and T. Pun, "MRML: Towards an extensible standard for multimedia querying and benchmarking – Draft proposal," Tech. Rep. 99.04, Computer Vision Group, Computing Centre, University of Geneva, rue Général Dufour, 24, CH-1211 Genève, Switzerland, October 1999.
`http://www.mrml.net/`
16. M. Beigi, A. B. Benitez, and S.-F. Chang, "Metaseek: A content-based meta-search engine for images," in *Symposium on Electronic Imaging: Multimedia Processing and Applications - Storage and Retrieval for Image and Video Databases VI, IST/SPIE'98, San Jose, CA*, 1998.
17. R. W. Picard, *Affective Computing*, MIT Press, Cambridge, 1997.
18. W. Müller, D. M. Squire, H. Müller, and T. Pun, "Hunting moving targets: an extension to Bayesian methods in multimedia databases," in Panchanathan *et al.*.[23] (SPIE Symposium on Voice, Video and Data Communications).
19. C. S. Lee, W.-Y. Ma, and H. Zhang, "Information Embedding Based on User's Relevance Feedback for Image Retrieval," in Panchanathan *et al.*.[23] (SPIE Symposium on Voice, Video and Data Communications).
20. C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Region-based image querying," in *Proceedings of the 1997 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, IEEE Computer Society, (San Juan, Puerto Rico), June 1997.
21. H. Freeman, "On the encoding of arbitrary geometric configurations," *IRE Trans. on Electronic Computers* **EC**(10), pp. 260–268, 1961.
22. H. Müller, W. Müller, D. M. Squire, and T. Pun, "Performance evaluation in content-based image retrieval: Overview and proposals," Tech. Rep. 99.05, Computer Vision Group, Computing Centre, University of Geneva, rue Gnral Dufour, 24, CH-1211 Genve, Switzerland, December 1999.
23. S. Panchanathan, S.-F. Chang, and C.-C. J. Kuo, eds., *Multimedia Storage and Archiving Systems IV (VV02)*, vol. 3846 of *SPIE Proceedings*, (Boston, Massachusetts, USA), September 20–22 1999. (SPIE Symposium on Voice, Video and Data Communications).