UNIVERSITE DE GENEVE

CENTRE UNIVERSITAIRE
D'INFORMATIQUE
GROUPE VISION

Date: February 14, 2000
N° 00.02

TECHNICAL REPORT

VISION

MRML: A Communication Protocol for Content-Based Image Retrieval

Wolfgang Müller*, Henning Müller,
Stéphane Marchand-Maillet, Thierry Pun
Computer Vision Group, University of Geneva
CH-1211 Genève 4, Switzerland

David McG. Squire
Computer Science and Software Engineering
Monash University
Melbourne, Australia

Zoran Pečenović
LCAV and Ergonomics Group
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland

Christoph Giess
Medical and Biological Informatics
Deutsches Krebsforschungszentrum
Heidelberg, Germany

Arjen P. de Vries
Department of Computer Science
University of Twente
Twente, The Netherlands

Computer Vision Group
Computing Science Center, University of Geneva
24 rue du Général Dufour, CH - 1211 Geneva 4, Switzerland

e-mail: muellerh@cui.unige.ch pun@cui.unige.ch

* viper@cui.unige.ch

Abstract

In this paper we introduce and describe the Multimedia Retrieval Markup Language (MRML). This XML-based markup language is the basis for an open communication protocol for content-based image retrieval systems (CBIRSs). MRML was initially designed as a means of separating CBIR engines from their user interfaces. It is, however, also extensible as the basis for standardized performance evaluation procedures.

Such a tool is essential for the formulation and implementation of common benchmarks for CBIR. A common protocol can also bring new dynamics to the CBIR field—it makes the development of new systems faster and more efficient, and opens the door of the CBIR research field to other disciplines such as Human-Computer Interaction. The MRML specifications, as well as the first MRML-compliant applications, are freely available and are introduced in this paper.

Keywords: Multimedia retrieval, Communication protocol, Evaluation framework, Reusable software components

1 Introduction

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 [3], while others need to be installed locally [13] and are specific to particular operating systems. The reuse of components in CBIR, *e.g.* user interfaces, is thus very rare. 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.* [2] 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 introduce the Multimedia Retrieval Markup Language (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 for multimedia databases. In other words, not only it is designed to fulfil 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 [14], which promoted the interoperability of medical imaging systems from different vendors. In summary, we address the urgent need for common tools which will facilitate the development and evaluation of multimedia database systems. By this means, we aim to facilite the development of common benchmarks for CBIRS performance, similar those used for textual information retrieval [17].

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. These are described briefly in § 2, and include two CBIR search engines (*Viper* and CIRCUS), which act as servers, and an interface (SnakeCharmer), which acts as a client. 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 [6], where the use of freely-available annotated image collections, such as [1], 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.

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*).

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 colour and texture features, both local and global).[1] Each image contains only a subset of these features. Access to images containing given features is provided by an inverted file, a standard access technique in text retrieval. The emphasis in *Viper* is on adapting the system response according to interaction with a user—positive and negative relevance feedback is accepted over several steps. Detailed descriptions of *Viper* may be found in [7, 16].

CIRCUS is a server framework supporting multiple image retrieval methods and algorithms.[2] Currently four methods are implemented. The first applies an adaptation of Latent Semantic Indexing [11] to image features describing local and global colour and texture, as well as global layout and optional keywords. The second is a texture/layout-specific method based on wavelet maxima moments. It extracts a set of contours from the image at various levels of detail, invariant to scale, translation, and partially to illumination changes. The third approach is texture-specific, it describes textures by computing the parameters of a Hidden Markov Model governing the coefficients of a wavelet decomposition of a textured image. The similarity is evaluated using the Kulbach-Leibler distance between two distributions. The last method is a fast, wavelet-packet, approximation of the Principal Component Analysis, based on the features used by the other methods. It is the most scalable and fastest of the implemented methods.

SnakeCharmer (figure 1) 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.

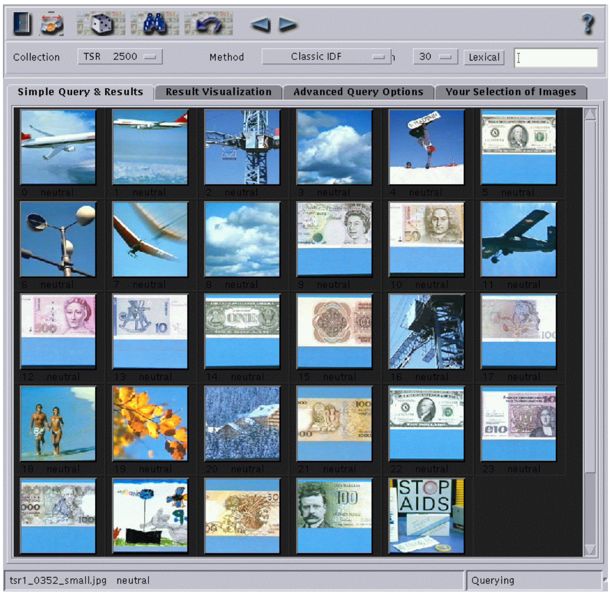

Figure 1: The JAVA interface SnakeCharmer

[1] http://viper.unige.ch/
[2] http://lcavwww.epfl.ch/CIRCUS

3 Multimedia Retrieval Markup Language

MRML is formally specified in [8]. It provides a framework which separates query formulation from 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 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:

- request of a capability description from 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 configuration of meta-queries during run time,
- formulation of QBE queries,
- transmission of user interaction data.

The final feature reflects our strong belief that affective computing [12] will soon play a role in the field of content-based multimedia retrieval. MRML already supports this by allowing the logging of some user interaction data. In particular, this is the case for the history-forward and history-backward functionalities of the SnakeCharmer interface.

Why XML and not CORBA? 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. Programming and debugging issues aside, 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, 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 Query Language). XML is about to become the main description language for all kinds of meta data on the internet and may also be used in MPEG-7 [5], thus ensuring the long-term support of its specifications.

Graceful degradation: independent development on a common base Graceful degradation is the key to successful independent extension of MRML. The basic principles can be summarised 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 § 4).

These principles provide guidelines for independent extensions of MRML. To avoid conflicts between differing extensions of MRML, 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.

3.2 Logging onto a CBIR 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. Skipping standard XML headers, the MRML code looks like this:

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

The server then informs the client of its capabilities. This message is empty in the current version of MRML, but it allows for the extension of the protocol:

```
<mrml>
 <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 then 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.

3.3 Interface configuration

The client can then request property sheet descriptions from the server. Different 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 [15] and a variety of methods for, and levels of, pruning. All these parameters are irrelevant for CIRCUS. 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. The interested reader is referred to [8] for details.

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* [9] or the system described by Lee *et al.* [4]. 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 order to demonstrate how easily MRML can be extended to other query paradigms, we give as an example QBE for images with user annotation. We assume that the user is invited to associate textual comments with images he or she marks as relevant or irrelevant. Since a tag for this purpose does not yet exist in MRML, we add an attribute `cui-user-annotation="..."` to the element. The prefix `cui-` is added to avoid name clashes with extensions from other groups which use MRML.

```
 <user-relevance-list>
  <user-relevance-element
   image-location="file:/images/1.jpg"
   user-relevance="1"
   cui-user-annotation="tropical fish"/>
 </user-relevance-list>
```

It is important to note here that servers which do not recognize the `cui-user-annotation` attribute still can make use of the remaining information contained in the `user-relevance-element` element.

As an example of how *not* to extend MRML, we give an extension with the same semantics but which does not respect the principle of graceful degradation:

```
 <user-relevance-list>
  <cui-user-relevance-element
   image-location="file:/images/1.jpg"
   user-relevance="1"
   user-annotation="tropical fish">
 </user-relevance-list>
```

Instead of adding an *attribute* to an existing MRML element (`user-relevance-element`), a new *element* was defined that contained the same kind of extension, namely `cui-user-relevance-element`. Consequently, servers which do not recognize this element will not be able to exploit any relevance information.

5 Conclusion

The development of MRML and the first MRML-compliant tools has 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. This should help 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. This new context will make it easy to develop common benchmarks for all MRML-compliant systems, similar to those which exist in the database and information retrieval communities. Only preliminary steps have been taken by the CBIR community towards developing common benchmarks—a comparison of evaluation techniques may be found in [6]. We are currently working on a more profound and flexible benchmarking system extending the results of this research.

Acknowledgements

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

References

[1] Annotated groundtruth database. Department of Computer Science and Engineering, University of Washington, web page:
`http://www.cs.washington.edu/ research/imagedatabase/groundtruth/`, 1999.

[2] Y.-C. Chang, L. Bergmann, J. R. Smith, and C.-S. Li. Query taxonomy of multimedia databases. In Panchanathan et al. [10]. (SPIE Symposium on Voice, Video and Data Communications).

[3] Surfimage webdemo. web page: `http://www-rocq.inria.fr/cgi-bin/ imedia/surfimage.cgi`, 1999.

[4] C. S. Lee, W.-Y. Ma, and H. Zhang. Information Embedding Based on User's Relevance Feedback for Image Retrieval. In Panchanathan et al. [10]. (SPIE Symposium on Voice, Video and Data Communications).

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

[6] H. Müller, W. Müller, D. M. Squire, and T. Pun. 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, dec 1999.

[7] 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. [10]. (SPIE Symposium on Voice, Video and Data Communications).

[8] 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. Technical Report 99.04, Computer Vision Group, Computing Centre, University of Geneva, rue Général Dufour, 24, CH-1211 Genève, Switzerland, October 1999.

[9] 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. [10]. (SPIE Symposium on Voice, Video and Data Communications).

[10] S. Panchanathan, S.-F. Chang, and C.-C. J. Kuo, editors. *Multimedia Storage and Archiving Systems IV (VV02)*, volume 3846 of *SPIE Proceedings*, Boston, Massachusetts, USA, September 20–22 1999. (SPIE Symposium on Voice, Video and Data Communications).

[11] Z. Pečenović. Image retrieval using latent semantic indexing. Final year graduate thesis, AudioVisual Communications Lab, Ecole Polytechnique Fédérale de Lausanne, Switzerland, June 1997.

[12] R. W. Picard. *Affective Computing*. MIT Press, Cambridge, 1997.

[13] QBIC$^{\text{TM}}$ – IBM's Query By Image Content. web page: `http://wwwqbic.almaden.ibm.com/~qbic/`, 1998.

[14] B. Revet. *DICOM Cook Book for Implementations in Madalities*. Philips Medical Systems, Eindhoven, Netherlands, 1997.

[15] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Department of Computer Science, Cornell University, Ithaca, New York 14853-7501, November 1987.

[16] 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)*, pages 143–149, Kangerlussuaq, Greenland, June 7–11 1999.

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