A Framework for Visual Information Retrieval

Horst Eidenberger[1], Christian Breiteneder[1], and Martin Hitz[2]

[1] Vienna University of Technology, Institute of Software Technology and Interactive Systems,
Favoritenstrasse 9-11 – 188/2, A-1040 Vienna, Austria
{eidenberger, breiteneder}@ims.tuwien.ac.at
[2] University of Klagenfurt, Department for Informatics-Systems, Universitätsstrasse 65-67,
A-9020 Klagenfurt, Austria
martin.hitz@uni-klu.ac.at

Abstract. In this paper a visual information retrieval project (VizIR) is presented. The goal of the project is the implementation of an open Content-based Visual Retrieval (CBVR) prototype as basis for further research on the major problems of CBVR. The motivation behind VizIR is: an open platform would make research (especially for smaller institutions) easier and more efficient. The intention of this paper is to let interested researchers know about VizIR's existence and design as well as to invite them to take part in the design and implementation process of this open project. The authors describe the goals of the VizIR project, the intended design of the framework and major implementation issues. The latter includes a sketch on the advantages and drawbacks of the existing cross-platform media processing frameworks: Java Media Framework, OpenML and Microsoft's DirectX (DirectShow).

1 Introduction

Global integration of information systems with the ability for easy creation and digitization of visual content have lead to the problem of how to manage these vast amounts of data in collections or databases. One of the crucial success factors of all approaches to this problem is apparently the implementation of effective but still easy to handle retrieval methods. Content-based retrieval of images and video (CBVR) is still a rather new approach to overcome these problems by deriving features (or: descriptors; like color histograms, etc.) from the visual content and comparing visual objects by measuring the distance of features with distance functions. CBVR can be a helpful addition to text retrieval systems. Its major advantages are fully automated indexing and description of visual content by visual features. On the other hand the fundamental drawbacks of this approach are:

- The semantic gap between high level concepts presented to a user and the low level features that are actually used for querying [22].
- Subjectivity of human perception. Different persons or the same person in different situations may judge visual content differently. This problem occurs in various situations: different persons may judge features (color, texture, etc.) differently, or if they judge them in the same way they still may perceive them in different ways [23].

Partly because of these two principle drawbacks four major problems of CBVR approaches can be identified:

- Low result quality—Using only general features for all types of visual content and asking the user to choose features her- or himself leads to retrieval results of low quality.
- Complicated interfaces—Casual users are overtaxed by the demand for a definite opinion on similarity, the selection of features and especially, by the often necessary provision of weights. Many users would not even try a typical CBVR interface, if they had the opportunity to use it. To improve the acceptance of CBVR systems simpler user interfaces are needed.
- Unsatisfactory querying performance—CBVR systems use distance functions to calculate the dissimilarity between visual objects. This process is often very slow and reply times in the range of minutes may occur for large databases.
- Lack of assessment methods—No standardized collections of images or videos exist for most types of features that could be used to assess new querying methods. One exception is the Brodatz database for textures, which is some sort of de-facto standard.

In this paper we present our visual information retrieval project (VizIR). The goal of this project is an open CBVR prototype as a basis for further research to overcome the problems pointed out above and in many other publications. VizIR was started in summer 2001 as a conclusion to the authors experiences with earlier CBVR projects and is currently evaluated for scientific funding in Austria. The motivation behind VizIR is: an open CBVR platform would make research (especially for smaller institutions) easier and more efficient (because of standardized evaluation sets and measures, etc.). Partly the authors took this idea from a panel discussion at the ACM Multimedia conference 2000 on a global multimedia curriculum, where the discussion participants stressed the need for shared scientific multimedia software. The intention of this paper is to let interested researchers know about VizIR's existence and design as well as to invite them to take part in the design and implementation process of this truly open project.

The rest of this paper is organized as follows: the following section points out relevant related work, section 3 is dedicated to the VizIR project goals, section 4 to the framework design and section 5 discusses major implementation issues.

2 Related work

Past CBVR research efforts have lead to several general-purpose prototypes like QBIC [8], VisualSEEk [26], Photobook [20], MARS and El Ninó [24] for image querying and OVID [16] or VIQS for video indexing and retrieval and some application specific prototypes like image retrieval systems for trademarks [28] or CueVideo for news videos analysis (e. g. [5]). These prototypes share a number of serious drawbacks:

- All of them implement only a small number of features and offer the developer no API for extension. An exception is IBM's QBIC system for image querying, which has (in version 3) a well-documented API for feature programming.
- Due to several reasons most prototypes are not available for further research. Some

of them have been canceled (e. g. Virage) and others have not been released to the public (e. g. Photobook).
- None of these prototypes have an architecture supporting the MPEG-7 standard (see [14]). To the knowledge of the authors at present no MPEG-7 compliant prototype for CBVR exists or is under development. Part 6 of MPEG-7 contains a reference implementation of its visual descriptors and a simple querying application, which was developed for testing and simulation [14]. Because it contains no framework, no documentation of the CBVR part, no user interface, no suitable database, no optimized descriptor extraction functions and no performance optimized algorithms unfortunately this reference implementation cannot be used as a CBVR prototype, although it is still a good starting point for developing one.

Apart from the mentioned focal points of research and the implemented prototypes the following key issues of CBVR systems have hardly been discussed so far:

- Similarity definition — The common way of similarity definition in CBVR systems is measuring distances with an L1 or L2 metrics (city block distance and Euclidean distance), merging a single objects distance values for multiple features by the weighted sum and presenting the user the objects with the lowest distance sum as the similar ones. In their publications the authors have shown that this method is far from being the most effective one [3]. More sophisticated methods for similarity definition would result in a qualitative better outcome (e. g. [25]).
- Media sets for assessment — As pointed out above, no considerable effort has been undertaken so far to put together standardized rated image and video sets for the various groups of features. This has lead to vague, often worthless statements on the quality of CBVR prototypes.
- Integration of computer vision methods — Surprisingly few ideas and methods have been taken over from the computer vision community up to now. Neural networks have been used for face detection and thresholding methods for segmentation but hardly any shaping techniques for 3D object reconstruction or sophisticated neural networks for scene analysis have been applied.

The VizIR project intends to integrate the various directions of past and current research in an open prototype to push CBVR research one step further towards practical usefulness by overcoming its most serious problems. The next chapter gives an overview of the objectives of the VizIR project.

3 Project goals

The VizIR project aims at the following major goals:

- Implementation of a modern, open class framework for content-based retrieval of visual information as basis for further research on successful methods for automated information extraction from images and video streams, definition of similarity measures that can be applied to approximate human similarity judgment and new, better concepts for the user interface aspect of visual information retrieval, particularly for human-machine-interaction for query definition and refinement and video handling.
- Implementation of a working prototype system that is fully based on the visual part

of the MPEG-7 standard for multimedia content description. Obtaining this goal requires the careful design of the database structure and an extendible class framework as well as seeking for suitable extensions and supplementations of the MPEG-7 standard by additional descriptors and descriptor schemes, mathematical and logical fitting distance measures for all descriptors (distance measures are not defined in the standard) and defining an appropriate and flexible model for similarity definition. MPEG-7 is not information retrieval specific. One goal of this project is to apply the definitions of the standard to visual information retrieval problems.

- Development of integrated, general-purpose user interfaces for visual information retrieval. Such user interfaces have to include a great variety of different properties: methods for query definition from examples or sketches, similarity definition by positioning of visual examples in 3D space, appropriate result display and refinement techniques and cognitively easy handling of visual content, especially video.
- Support of methods for distributed querying, storage and replication of visual information and features and methods for query acceleration. The importance of this issue becomes apparent from the large amount of data that has to be handled in such a system and the computation power that is necessary for querying by – often quite complex – distance functions. Methods for distributed querying, storage and replication include the replication of feature information, client-server architectures and remote method invocation in the querying and indexing modules as well as compression of video representations for the transport over low bandwidth networks. Methods for query acceleration include indexing schemes, mathematical methods for complexity reduction of distance functions and generation of querying heuristics [4].

Another implicit goal of the VizIR project is the development of a multimedia specific UML-based software development process. Multimedia applications have special needs that have to be considered during the design and implementation of such a system. Developing tailor-made software development methods on the basis of the UML design process is just a logical step. The next section will give technical details on these objectives and the intended system architecture.

4 Framework design

Referred to its nature the VizIR project can be split in a front-end part (user interfaces for query definition, result display and query refinement, video representation and delivery, etc.) and a back-end part (class framework for querying, information management, etc.).

The major issues concerning the front-end are:

- Design of image querying interfaces—Modern ways of similarity definition (3D spatial layout of example images, iconic indexing, etc.) have to be combined with different querying paradigms (query by example, query by sketch, etc.). Additionally, it must be possible to define spatial relationships within visual content, regions of interest, etc.

- Video presentation and interaction—Implementation of state-of-the-art video handling paradigms (e. g. micons, panoramas, paper video, etc.) and development of new, better metaphors. One interesting alternative could be a spatio-temporal onion view on video objects.
- Design of video querying interfaces—The video handling methods have to be integrated in a video querying interface, which has to offer analogous features to the image querying interface and additional query by (moving) objects methods.
- Integration of image and video querying—The media querying interfaces have to be integrated in a joint user interface where image features can be applied to video clips and videos consisting of different views on a scene for image querying.
- Application-specific interfaces—In addition to general- purpose interfaces methods have to be developed to adapt these interfaces to application specific needs. Fields of application in the future will be digital libraries, CSCW systems and of course the Internet itself.
- Result display interfaces—This is a rather easy task for images (e. g. browsing, iconic indexing, etc.) but hard to implement for video content. Common approaches are index frames and micons, which are obviously unsatisfactory. A more sophisticated approach could be an object viewer for all objects and their temporal trajectories in a video shot. Also, video cubism (interactively cutting an X-Y-time cube of video data along arbitrary oriented planes; [7]) should be considered as an alternative for offering video results.
- Query refinement interfaces — Iterative query refinement by relevance feedback is a technique that has become state-of-the-art in information retrieval applications in the last years [15], [27]. The effect of such a component stands and falls with an intuitive user interface that allows the user to enter his feedback in an intuitive way.

These interfaces have to be designed as intuitive and self-explanatory as possible to guarantee high usability and in consequence increasing acceptance for CBVR. Matters related to the design and implementation of the back-end are:

- Implementation of a technically sound class framework for the other system components. Even though this is not a research but an engineering problem, the authors have to stress that using a professional database and programming environment will be crucial success factors for a modern CBVR research prototype. As pointed out above, most past approaches have serious shortages in their system architecture. VizIR will use a professional relational database for media and feature data storage and an open class framework as basis for the implementation of further components.
- Implementation of the basic MPEG-7 descriptors for still images and video. It is intended to follow the reference implementation of part 6 of the standard. For the reasons given above and especially because the algorithms of the reference implementation are not optimized the redesign and implementation of the MPEG-7 descriptors is a very time- and human resources consuming task.
- The basic MPEG-7 descriptors can be combined with aggregate descriptors (grid layout, time line, etc.) and grouped to descriptor schemes. The task of this part of the project is to discuss, which combinations of descriptors make sense for a general-purpose CBVR prototype. Additionally, an API has to be defined for the creation of descriptor schemes.

- MPEG-7 is not a visual information retrieval specific standard and in general does not include distance functions for the various descriptors. Neither does it give any recommendations. Therefore it is necessary to implement common distance metrics (like L1, L2 metric, Mahalanobis distance, etc.; [23]), to associate them with descriptors and to develop custom distance functions where these metrics are not applicable (e. g. object features, etc.).
- The MPEG-7 standard – although it is a major advance in multimedia content description – standardizes only a number but not nearly all useful features. It is necessary to design and implement additional descriptors and distance functions for texture description of images (wavelets, etc.; e. g. [13]), symmetry detection of objects (useful for face detection, detection of human-made objects, etc.), object description in video streams (structure recognition from motion, etc.), object representation (scene graphs, etc.) and classic video analysis (shot detection, etc.) from uncompressed as well as compressed video streams. Additionally the authors plan to use fractal methods (iterated function systems; IFS) to describe the shape of objects effectively. So far IFS have been used for the compression of self-similar objects (e. g. [1]) but hardly for content-based retrieval (see [12]). The authors think, that IFS could be very effective for shape description too.
- Design of methods for query definition that are flexible enough to satisfy different ways of how humans can perceive and judge similarity which are still applicable in a distributed querying environment. The query model approach developed by the authors could be applied and extended for this task [3].
- Implementation of methods for query refinement. As frequently stressed in publications on information retrieval this is a crucial task for the quality of a retrieval system. VizIR will contain methods for experimenting with feedback by rating and positive query examples. The authors doubt that approaches with positive and negative query examples make sense for visual content.
- Development and implementation of indexing schemes and query acceleration models. Next to classic index structures for visual content (e. g. R tree, segment index tree, etc.) and query acceleration techniques (application of the triangle inequality [2], storage of the factorized terms of the Mahalanobis distance [21], etc.) experiments will be undertaken with new heuristic approaches like those previously published by the authors [4].
- Finally, it is necessary to implement tools for distributed and replicated visual content management as well as database management. This is – like the first element of this list – more an engineering than a research problem (except the feature replication problem).

A third group of matters, which is relevant to both the front-end and the back-end concerns assessment methods. To the belief of the authors a significant improvement of CBVR research in the future will be the development of standardized quality assessment procedures. In the VizIR project the following assessment tasks will be undertaken:

- Analysis of common evaluation models (recall, precision, etc.; [9], [19]) and application of other methods (systematic measures, etc.). Moreover different evaluation techniques and methods from other research areas will be checked for applicability to the problem at hand. This could be conventional psychological methods, e. g. semantic differential techniques [18] or new methods to be developed. The major problem - apart from the cumbersome lack of standardized evaluation sets - of applying the standard measures in information retrieval, recall and precision to CBVR systems that use linear weighted merging (see above) is that this implicitly means giving up at least 10% of recall. This is because a system with linear weighted merging returns the n “most similar” available objects (independent of the question whether or not they are really similar), while the recall measures the ratio of really similar objects to all available objects.

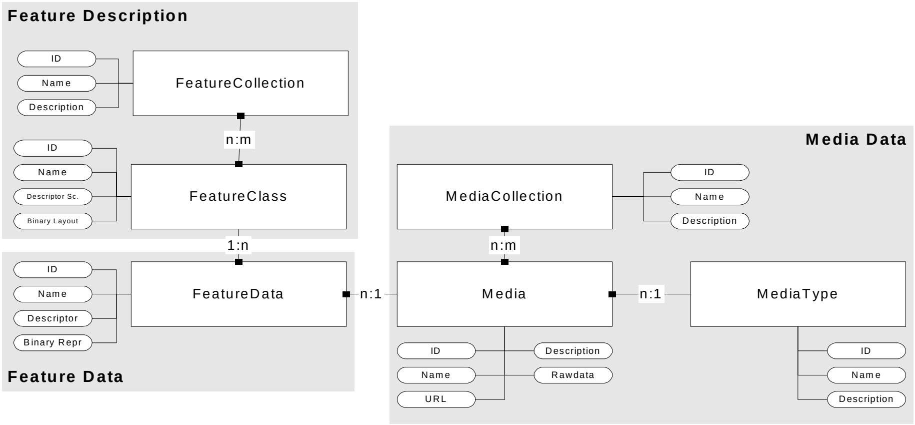

Fig. 1. EER database diagram. Visual media is stored in table “Media” and associated with a single “MediaType”. Each media may belong to n collections and each collection may contain m elements. Feature classes are described in table “FeatureClass” with the MPEG-7 descriptor definition language (DDL; based on XML schema). Features are organized in collections as well. Feature data is stored in binary and DDL format in table “FeatureData”.

- Creation of evaluation sets with image or video content for groups of descriptors and assignment of pair-wise similarity from tests with volunteers (students, etc.). Such sets are obviously decisive for the quality judgment of CBVR systems but in fact there is only one de-facto standard, the Brodatz database for texture images. The aim of the VizIR project is the definition of test sets for shape features, color and symmetry features and video object features. Partially these evaluation sets will be created by enriching and extending the image and video clip sets, which were used for building the ground truth of some MPEG-7 features (e. g. motion activity descriptor, etc.). Different approaches - e.g., findings on the basis of gestalt laws - will be checked for their suitability to develop those test sets.
- Extended evaluations on the MPEG-7 descriptors and descriptor schemes as well as on the other implemented descriptors and aggregates with statistical methods in two steps:
  - Evaluation of their independent performance and their performance in combinations. From this information the overall performance of the visual part of MPEG-7 and VizIR can be judged.
  - Analysis of dependencies among descriptors with statistical methods (cluster analysis, factor analysis, etc.) to identify a base for the space of descriptors and become able to normalize the visual part of the MPEG-7 standard and extend it by new independent descriptors.

- Evaluation of the performance optimization methods implemented in VizIR in comparison to other comparable retrieval systems.
- Finally assessment of the user interfaces by volunteers who judge the video handling methods, similarity definition concepts and the overall usability of the system. For this task methods of usability assessment will be applied.

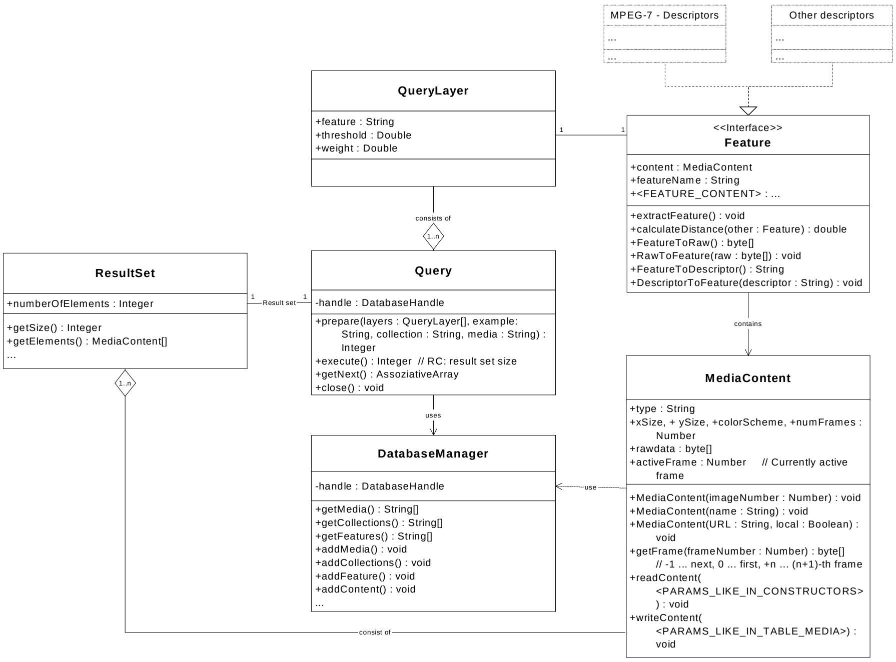
MPEG-7 - Descriptors
...
...
Other descriptors
...
...
QueryLayer
+feature : String
+threshold : Double
+weight : Double
1
1
<<Interface>>
Feature
+content : MediaContent
+featureName : String
+<FEATURE_CONTENT> : ...
+extractFeature() : void
+calculateDistance(other : Feature) : double
+FeatureToRaw() : byte[]
+RawToFeature(raw : byte[]) : void
+FeatureToDescriptor() : String
+DescriptorToFeature(descriptor : String) : void
consists of
1..n
ResultSet
+numberOfElements : Integer
+getSize() : Integer
+getElements() : MediaContent[]
...
1
Result set
1
Query
-handle : DatabaseHandle
+prepare(layers : QueryLayer[], example: String, collection : String, media : String) : Integer
+execute() : Integer // RC: result set size
+getNext() : AssoziativeArray
+close() : void
contains
MediaContent
+type : String
+xSize, + ySize, +colorScheme, +numFrames : Number
+rawdata : byte[]
+activeFrame : Number // Currently active frame
+MediaContent(imageNumber : Number) : void
+MediaContent(name : String) : void
+MediaContent(URL : String, local : Boolean) : void
+getFrame(frameNumber : Number) : byte[]
// -1 ... next, 0 ... first, +n ... (n+1)-th frame
+readContent(<PARAMS_LIKE_IN_CONSTRUCTORS>) : void
+writeContent(<PARAMS_LIKE_IN_TABLE_MEDIA>) : void
1..n
uses
DatabaseManager
-handle : DatabaseHandle
+getMedia() : String[]
+getCollections() : String[]
+getFeatures() : String[]
+addMedia() : void
+addCollections() : void
+addFeature() : void
+addContent() : void
...
use
consist of

Fig. 2. UML class diagram for an ideal implementation of the VizIR class framework. Key element is class “Query”, which contains the methods for query generation and execution. Each query consists of a number of “QueryLayer” elements that implement exactly one feature each. All feature classes – MPEG-7 descriptors as well as all others - are derived from the interface “Feature” and contain methods for descriptor extraction (“extractFeature()”), serialization (“FeatureToRaw()”, “RawToFeature()”, etc.) and distance measurement (“calculateDistance()”). Feature classes take their media content from instances of the class “MediaContent”. The result of each query is a set of media objects (represented as MediaContent objects), which is stored in a “ResultSet” object. Finally the methods of class “DatabaseManager” encapsulate the database access.

The latter two evaluation cycles have to be performed in usability labs. A combination of different observation methods and devices - such as eye-trackers and video observation devices – is necessary to collect objective (e.g. eye-movement) as well as subjective data (e.g. verbal expressions). By analyzing and comparing different data, cost and benefit assessments of existing systems with special focus on the system to be developed are possible.

The VizIR prototype will be based on a standard relational database. Fig. 1 gives an overview of its tables and relations for media and feature storage. Fig. 2 outlines the likely class structure of the VizIR prototype. To a certain extent this class framework follows the architecture of IBM’s QBIC system [8], but largely differs from QBIC in its server/client independent classes. Similarly to QBIC, the database access is hidden from the feature programmer and the layout of all feature classes is predefined by the interface "Feature".

Concluding this sketch of the VizIR prototypes system architecture we outline several aspects of the application and data distribution. Modern CORBA based programming environments like the Java environment permit the network-independent distribution of applications, objects and methods (in Java through the Remote Method Invocation library) to increase the performance of an application by load balancing and multi-threading. If VizIR will be implemented in Java the objects for querying could be implemented as JavaBeans, feature extraction functions with RMI, database management through servlets and user interfaces as applets. Database distribution could be realized through standard replication mechanisms and database access through JDBC.

5 Implementation

The major question concerning the implementation of the VizIR prototype is on the programming environment. At this point in time when MPEG-21 is still far out of sight, there are three major alternatives that support image and video processing to choose from:

- Java and the Java Media Framework (JMF; [10])
- The emerging Open Media Library standard (OpenML) of the Khronos group [17]
- Microsoft DirectX (namely DirectShow) resp. its successor in the .NET environment [6]

All of these environments offer comprehensive video processing capabilities and are based on modern, object-oriented programming paradigms. DirectX is platform-dependent and a commercial product. For .NET Microsoft has recently initiated the development a Linux version but it is expected that this version will not be available before summer 2002 and will still have to be purchased. Additionally it is unlikely that versions for other operating systems will be developed as well (SunOS, OpenBSD, IRIX, etc.). Therefore in the following discussion we will concentrate on the first two alternatives: JMF and OpenML. JMF is a platform-dependent add-on to the Java SDK, which is currently available for SunOS and Windows (implementation by SUN and IBM) as well as Linux (implementation by Blackdown) in a full version and in a Java version with less features for all other operating systems that have Java Virtual Machine implementations. JMF is free and extensible. OpenML is an initiative of the Khronos Group (a consortium of companies with expert knowledge in video processing, including Intel, SGI and SUN) that standardizes a C-interface for multimedia production. OpenML includes OpenGL for 3D and 2D vector graphics, extensions to OpenGL for synchronization, the MLdc library for video and audio rendering and the 'OpenML core' for media processing (confusingly the media processing part of OpenML is named OpenML as well; therefore we will use the term 'OpenML-mp' for the media processing capabilities below). The first reference implementation of OpenML for Windows was announced for winter 2001.

Among the concepts that are implemented similarly in JMF and OpenML-mp are the following:

- Synchronization: a media objects time base (JMF: TimeBase object, OpenML-mp: Media Stream Counter) is derived from a single global time base (JMF: SystemTimeBase object, OpenML-mp: Unadjusted System Time)
- Streaming: both environments do not manipulate media data as a continuous stream but instead as discrete segments in buffer elements.
- Processing control: JMF uses Control objects and OpenML-mp uses messages for this purpose.

Other important media processing concepts are implemented different in JMF and OpenML-mp:

- Processing chains: in JMF real processing chains with parallel processing can be defined (one instance for one media track is called a CodecChain). In OpenML-mp processing operations data always flows from the application to a single processor (called a Transcoder) through a pipe and back.
- Data flow: JMF distinguishes between data sources (including capture devices, RTP servers and files) and data sinks. OpenML-mp handles all I/O devices in the same way (called Jacks).

The major advantages of OpenML-mp are:

- Integration of OpenGL, the platform-independent open standard for 3D graphics.
- A low-level C API that will probably be supported by the decisive video hardware manufacturers and should have a superior processing performance.
- The rendering engine of OpenML (MLdc) seems to have a more elaborate design than the JMF Renderer components. Especially it can be expected that the genlock-mechanism of MLdc will prevent lost-sync phenomena, which usually occur in JMF when rendering media content with audio and video tracks that are longer than ten minutes.
- OpenML-mp defines more parameters for video formats and is closer related to professional video formats (DVCPRO, D1, etc.) and television formats (NSTC, PAL, HDTV, etc.)

On the other hand the major disadvantages of OpenML are:

- It is not embedded in a CASE environment like Java for JMF. Therefore application development requires more resources and longer development cycles.
- OpenML is not object-oriented and includes no mechanism for parallel media processing.

The major drawbacks of JMF are:

- Lower processing performance because of the high-level architecture of the Java Virtual Machine. This can be reduced by the integration of native C code through the Java Native Interface.
- Limited video hardware and video format support: JMF has problems with accessing certain video codecs, capture devices and with transcoding of some video formats.

The outstanding features of JMF are:

- Full Java integration. The Java SDK includes comprehensive methods for distributed and parallel programming, database access and I/O processing. Additionally professional CASE tools exist for software engineering with Java.
- JMF is free software and reference implementations exist for a number of operating systems. JMF version 2.0 is a co-production of SUN and IBM. In version

1.0 Intel was involved as well.

- JMF is extensible. Additional codecs, multiplexers and other components can be added by the application programmer.

The major demands for the VizIR project are the need for a free and bug-free media processing environment that supports distributed software engineering and has a distinct and robust structure. Matters like processing performance and extended hardware support are secondary for this project. Therefore the authors think that currently JMF is the right choice for the implementation.

Design and implementation will follow a UML based incremental design process and prototyping, because UML is state-of-the art in engineering and because of the valuable positive effect of rapid prototyping on the employee's motivation. Standard statistical packages and Perl scripts will be used for performance evaluation and Self-organizing Maps [11] and Advanced Resonance Theory (ART) neural networks as well as genetic algorithms for tasks like pattern matching and (heuristic) optimization (like in [4]).

6 Conclusion

The major outcome of the open VizIR project can be summarized as follows:

- An open class framework of methods for feature extraction, distance calculation, user interface components and querying.
- Evaluated user interfaces methods for content-based visual retrieval.
- A system prototype for the refinement of the basic methods and interface paradigms.
- Carefully selected evaluation sets for groups of features (color, texture, shape, motion, etc.) with human-rated co-similarity values.
- Evaluation results for the methods of the MPEG-7 standard, the authors earlier content-based retrieval projects and all other promising methods.

The authors would like to invite interested research institutions to join the discussion and participate in the design and implementation of the open VizIR project.

References

1. Barnsley, M.F., Hurd, L.P., Gustavus, M.A.: Fractal video compression. Proc. of IEEE Computer Society International Conference, Compcon Spring (1992)
2. Barros, J., French, J., Martin, W.: Using the triangle inequality to reduce the number of comparisons required for similarity based retrieval. SPIE Transactions (1996)
3. Breiteneder, C., Eidenberger, H.: Automatic Query Generation for Content-based Image Retrieval. Proc. of IEEE Multimedia Conference, New York (2000)
4. Breiteneder, C., Eidenberger, H.: Performance-optimized feature ordering for Content-based Image Retrieval. Proc. European Signal Processing Conference, Tampere (2000)
5. Chua, T., Ruan, L.: A Video Retrieval and Sequencing System. ACM Transactions on Information Systems, Vol. 13, No. 4 (1995) 373-407

6. DirectX: msdn.microsoft.com/library/default.asp?url=/library/en-us/wcegmm/htm/dshow.asp
7. Fels, S., Mase, K.: Interactive Video Cubism. Proc. of ACM International Conference on Information and Knowledge Management, Kansas City (1999) 78-82
8. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by Image and Video Content: The QBIC System. IEEE Computer (1995)
9. Frei, H., Meienberg, S., Schäuble, P.: The Perils of Interpreting Recall and Precision. In: Fuhr, N. (ed.): Information Retrieval, Springer, Berlin (1991) 1-10
10. Java Media Framework Home Page: java.sun.com/products/java-media/jmf/index.html
11. Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J.: SOM-PAK: The Self-organizing Map Program Package. Helsinki (1995)
12. Lasfar, A., Mouline, S., Aboutajdine, D., Cherifi, H.: Content-Based Retrieval in Fractal Coded Image Databases. Proc. of Visual Information and Information Systems Conference, Amsterdam (1999)
13. Lin, F., Picard, R. W.: Periodicity, directionality, and randomness: Wold features for image modelling and retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence (1996)
14. MPEG-7 standard: working papers www.cselt.it/mpeg/working_documents.htm#mpeg-7
15. Nastar, C., Mitschke, M., Meilhac, C.: Efficient Query Refinement for Image Retrieval. Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (1998)
16. Oomoto, E., Tanaka, K.: OVID: design and implementation of a video-object database system. IEEE Transactions on Knowledge and Data Engineering (1993)
17. OpenML: www.khronos.org/frameset.htm
18. Osgood, C. E. et al.: The Measurement of Meaning. University of Illinois, Urbana (1971)
19. Payne, J. S., Hepplewhite, L., Stonham, T. J.: Evaluating content-based image retrieval techniques using perceptually based metrics. SPIE Proc., Vol. 3647 (1999) 122-133
20. Pentland, A., Picard, R. W., Sclaroff, S.: Photobook: Content-Based Manipulation of Image Databases. SPIE Storage and Retrieval Image and Video Databases II (1994)
21. Rui, Y., Huang, T., Chang, S.: Image Retrieval: Past, Present and Future. Proc. of International Symposium on Multimedia Information Processing, Taiwan (1997)
22. Santini, S., Jain, R.: Beyond Query By Example. ACM Multimedia (1998)
23. Santini, S., Jain, R.: Similarity Measures. IEEE Transactions on Pattern Analysis and Machine Intelligence (1999)
24. Santini, S., Jain, R.: Integrated browsing and querying for image databases. IEEE Multimedia, Vol. 3, Nr. 7 (2000) 26-39
25. Sheikholeslami, G., Chang, W., Zhang, A.: Semantic Clustering and Querying on Heterogeneous Features for Visual Data. ACM Multimedia (1998)
26. Smith, J. R., Chang, S.: VisualSEEk: a fully automated content-based image query system. ACM Multimedia (1996)
27. Wood, M., Campbell, N., Thomas, B.: Iterative Refinement by Relevance Feedback in Content-Based Digital Image Retrieval. ACM Multimedia (1998)
28. Wu, J. K., Lam, C. P., Mehtre, B. M., Gao, Y. J., Desai Narasimhalu, A.: Content-Based Retrieval for Trademark Registration. Multimedia Tools and Applications, Vol. 3, No. 3 (1996) 245-267