Format independence of audio and video in multimedia database systems

Maciej Suchomski*
and Klaus Meyer-Wegener

Department of Computer Science,
Friedrich-Alexander University of Erlangen-Nuremberg,
Martensstr. 3, 91058 Erlangen, Germany
E-mail: Maciej.Suchomski@informatik.uni-erlangen.de
E-mail: Klaus.Meyer-Wegener@informatik.uni-erlangen.de
*Corresponding author

Abstract: Format independence is a crucial aspect of data independence in databases, but it is rarely offered for continuous media data (such as audio and video) in multimedia databases. It can be provided by a real-time transformation supported by prepared meta-data, which is most suitable for systems with a low frequency of use for certain continuous multimedia data and a high probability of differentiation between clients accessing these data. This paper focuses on such a real-time transformation that is based on the RETAVIC framework. Additionally, the application scenario and requirements for long-term internal storage formats are discussed.

Keywords: format independence; transcoding; real-time conversion; multimedia databases; format adaptation; audio; video; continuous data; meta-data based conversion; data transformation.

Reference to this paper should be made as follows: Suchomski, M. and Meyer-Wegener, K. (2008) 'Format independence of audio and video in multimedia database systems', *Int. J. Intelligent Information and Database Systems*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: M. Suchomski is a PhD candidate at the Database Systems Chair. He received MSc in Computer Science from Wroclaw University of Technology in Poland in 2001. His current scientific work concerns format independence provision of continuous data for multimedia database management systems, multimedia servers and meta-data based processing.

K. Meyer-Wegener is a full Professor and Head of the Database Systems Chair since 2001. He received his PhD in Computer Science from the University of Kaiserslautern in Germany. He finished his habilitation also at the University of Kaiserslautern in 1991 and was a full Professor at the Department of Computer Science of Dresden University of Technology in Germany for more than eight years. His current research covers multimedia databases, document and workflow management and digital libraries.

1 Introduction

In this day and age, multimedia data find an application in many different areas on a number of various platforms. The consistently raising interest in consumption of multimedia by usual people is followed by rising attractiveness of the potential market for the information and entertainment providers. New standards and solutions have emerged and different fields of application have been found for media data.

Progress in the research on digital wired network technologies, wireless LAN and terrestrial digital broadcasting networks resulted in raised internet throughput delivering higher quality of media data to end users. On the other hand, despite the very active evolution of cellular networks resulting in 3GG, such networks have limited and erroneous throughput capabilities.

In parallel, huge efforts put on the compression algorithms resulted in new and very efficient implementations of the audio and video codecs. A broad cooperation of science and industry on AV coding algorithms resulted in the stabilisation of many standards, e.g., ITU-T H.264/AVC (2005), MPEG-4 Part 2 (2001) and MPEG-4 SLS (Geiger and Yu, 2005; Suchomski et al., 2006).

However, there still is a huge gap between these standards and the appropriate solution for managing multimedia data. The researchers working in the database field do not follow the current market trend in respect to the multimedia data. They rather focus on other areas like data streams and sensor networks, data integration, process and workflow management, data warehouses, or repositories. The multimedia area is neglected, but there is still a lot to investigate, even in multimedia query languages (Melton and Eisenberg, 2001).

One of the basic ideas of the databases, which has driven the researchers for many years, namely providing the data independence, is cleverly omitted in respect to the multimedia data (continuous, timed data). It is especially noticeable when talking about format independence, which means that users access the data without referring to the storage format, and the DBMS is able to return them in many different formats and qualities, and therefore, the delivered format can be chosen at will independently from the internal storage format. Of course, a database management system is expected to support a large variety of different applications, thus it hides the storage format and delivers the data in the format understood by the end-user application or by the middleware. However, this is not the case for multimedia data. Here, the possibility of selection of the required multimedia data regardless of the differences between the storage and delivery format is not given to the end users. An option to choose between lossless and lossy data, higher or lower quality with selection of the delivery format, which is in some cases very important especially in the scientific world, is still a dream.

It could be provided by the transparent transformation of media data, but this is not an easy task, because of the characteristics of multimedia data (Suchomski et al., 2004) as well as because of the process of transformation itself. To provide continuous delivery to the end-client with certain level of quality, any format conversion process of timed data requires both enormous computational power and real-time execution.

2 Related work

The currently available media servers have huge deficiencies with respect to format independence for the Media Objects (MOs). The systems we analysed and compared support only a small number of formats (Bente, 2004), not to mention the possibility to transform one video format into the other, when a format for the output video is given. The scalability of the quality is provided only (if at all) with respect to one specific format (i.e., media data are kept in some pre-processed instances with different properties). The only server allowing for limited transformation (from RealMedia Format to Advanced Streaming Format) supports neither QoS control nor RT processing.

When transforming one video format into another, recompression of the video data is required. To achieve reasonable processing speeds, modern video encoders (e.g., the popular XviD MPEG-4 video codec) employ sophisticated block-matching algorithms instead of a straightforward full search to reduce the complexity of Motion Estimation (ME). Often, predictive algorithms like EPZS (Tourapis, 2002) are used, which offer a 100–5000 times speed-up over a full search while achieving similar picture quality. The performance of predictive search algorithms, however, highly depends on the characteristics of the input video (and is especially low for sequences with irregular motion). This content-dependent and unpredictable behaviour of the ME step makes it very difficult to implement video encoders in real-time. However, this unpredictability can be eliminated without compromising compression efficiency by using Meta-Data (MD) (Suchomski et al., 2005).

Another approach is Universal Multimedia Access supported by MPEG-21 (2004). An overview on MPEG-21 Digital Item Adaptation is given by Vetro (2004). Here, it is not the goal to provide data independence (any format requested by application), but rather to adopt the existing scalable format to the network environment (e.g., recent work on MPEG-4 Scalable Video Coding). One can compare transcoding to scalable coding with adaptivity (Vetro, 2003), but in our perspective data independence is the driving force, so any adaptation of one universal format with different qualities is not considered as applicable.

There is also other work on MD-based approaches of video transcoding, e.g., in MPEG-7, there is the MediaTranscodingHints descriptor in the MediaProfile descriptor of Media Description Tools (MPEG-7, 2003). However, the properties proposed there do not fit the real encoding implementation in our opinion. Among others, they define the general motion, shape and coding hints, which cannot be used directly by the encoder. The goal here was to simplify the execution in general for example by limiting motion search range. Moreover, a property like 'intraFrameDistance' (MPEG-7, 2003) is not a good hint, since scene changes may appear unpredictably (and usually intra frames are used then). Regardless, few parameters are similar (e.g., importance is related to priority), and some very important ones are still missing (frame type, MB mode, etc.).

On the other hand, the research in the video-processing field discusses the specific transcoding problems in more detail (Vetro et al., 2003; Sun et al., 2005) and especially the applications of an MD-based approach (Vetro et al., 2000; Suzuki and Kuhn, 2000; Kuhn and Suzuki, 2001). Here, however the focus is on the transcoding process itself and does not consider real-time processing and QoS control specific for the MMDBMS in broader context.

The only example of an end-to-end video streaming and transcoding system, to our knowledge, is discussed in Sun et al. (2005). There are few elements common with our architecture pointed out, but also few critical are missing (content analysis, MD-based encoding). Moreover, the discussed test-bed architecture is FGS-based, which is also one of the adaptation solutions. The extensions to MPEG-4 transcoding of the test bed are also proposed, but it assumes that there are several uncompressed resolutions (which is not our goal) and still do not consider the application of MD-based encoding algorithms.

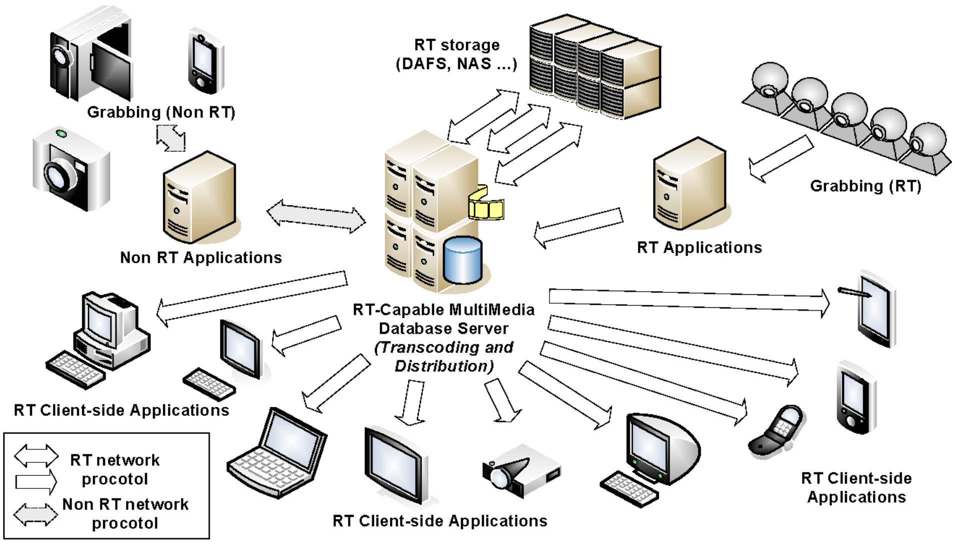
In our media transformation framework (Suchomski et al., 2005), each media-object is processed at least twice before being delivered to a client, and media transformations are assisted by MD. This is analogical to two-pass encoding techniques in video compression (Westerink et al., 1999), so the optimisation techniques deriving from the two-pass idea can also be applied to our approach. However, our framework goes beyond that – it heavily extends the idea of MD-assisted processing and employs MD to reduce the complexity and enhance the predictability of the transformations to meet real-time requirements.

3 Application scenario and requirements

We envision a scenario (Figure 1) where many different output formats for different platforms (laptops, PDAs, cell phones, desktops, multimedia workstations, mobile devices, tablets, or even digital cinemas with CRT projectors, etc.) can be generated from a single media-object storage format (Suchomski et al., 2005). Please note that this is not a video-on-demand scenario where there are very large numbers of users with very similar environments, i.e., TV sets with set-top boxes. Neither is it an internet music store scenario where plenty of clients use the same software for accessing the data, e.g., iTunes and QuickTime Player. Moreover, it is not intended for video streaming in the internet where a limited number of players are used in the context of web browsers. The user group in our case is relatively small, but it issues extremely varying requests for the same media data not necessarily corresponding to well-defined standards (e.g., scientists or managers). So, the MOs cannot be stored redundantly in all the formats needed, the users can be limited neither to only a few formats nor to one scalable format. Our scenario also assumes that clients access the data on-demand and just-in-time, which causes specific real-time delivery constraints and thus requires QoS control. Additionally, we do not restrict insertion of data, neither to a real-time nor to a non-real-time application (grabbing or editing), but we assume that the data before delivery is analysed and stored in the internal storage format.

The essential objective in the Real-Time Audio-Video Conversion (RETAVIC) project (Suchomski et al., 2005) is to develop a functionality extending nowadays' multimedia database services that brings out efficient and format-transparent access to multimedia data by utilising real-time format conversion respecting user's specific demands. This of course covers not only end-user demands, but also other frameworks like Multimedia For You (MM4U) (Boll, 2005), where the user-centric perspective of multimedia delivery is the main assumption, and the adaptation to end-device-specific capabilities is carried out during the composition. The delivery techniques like transcoding, content-adaptation and adaptive streaming are crucial in the user-centred MM-content distribution, and parameters like display size, presentation software and decoding capabilities pose specific challenges to audio and video retrieval (Scherp and Boll, 2005).

Figure 1 Various clients scenario using very large MMDB (see online version for colours)



In respect to applications and their needs for format independence, two aspects must be considered. At first, one should choose between an on-the-fly (or online) and offline provision of the requested data. The first case is considered by us and it is chosen as the goal of the RETAVIC project – in our opinion, a database management system that will deliver the suitable data, e.g., two hours after obtaining the request, would be hardly usable in reality.

The second aspect of delivering format independence of the data must consider the information completeness that is crucial in the long-time storage. Users may require media data to keep all the information to analyse it in the later time. If audio or video data are compressed, some information is usually lost due to the lossy compression process (unless a lossless transformation is applied).

Moreover, the long-time storage of the (multimedia) data should not be limited to only one technology forever and should be allowed to utilise new technological achievements (e.g., developing a new, more efficient compression algorithms). This is only possible if the migration from the old format to a new one has been considered in the design. Of course, none of the information may be lost or changed during the migration, i.e., the tandem coding problems must be avoided.

Thus, the storage format considered by us must be a lossless format. On the other hand, the lossless formats cause some efficiency problems, because the lossless codecs are not designed for effective, scalable storage and retrieval as the primary goal (the objective is a mathematically invertible function causing no data losses in processing). Therefore, the efficiency issues of the data storage and access must be considered and a trade-off between compression efficiency and its complexity must be found, i.e., only fast, effective, and scalable (reflecting processing and access) methods providing lossless compression should be employed.

4 The core of the RETAVIC architecture

The foundations of the RETAVIC architecture (Suchomski et al., 2005) are two processing parts: a non real-time preparation phase and the real-time transformation phase. In the non-real-time preparation phase, the insertion of MOs into the media server is performed. Here, the integration of MOs is achieved by importing different formats and decoding them to raw format if necessary (format conversion module), and next, the part employs an encoding algorithm to produce a lossless and scalable internal storage format. The internal formats for the RETAVIC architecture have already been proposed, namely: Layered Lossless Video – LLV1 in short – (Militzer et al., 2005) for video and MPEG-4 SLS (Geiger and Yu, 2005) for audio. Both formats fulfil the requirements of being lossless and scalable. The second important aspect during insertion of the MOs is a content analysis of the data. This step is used to produce the MD required later in the real-time transcoding part.

The real-time transformation part provides format independence to the client applications of MMDBMS. It employs a media transcoding that meets real-time requirements. The converters use a pipelining scenario to process MOs, passing the so-called media quanta among them (Suchomski et al., 2004). Two classes of converters are defined here regardless of the media type (i.e., valid for both audio and video). One is the *real-time decoder* that transforms the internal layered format selectively into raw intermediate format. The second one is the *real-time encoder* that runs the compression algorithm requested by the user.

5 Achieving data independence by Meta-Data based transcoding

The multimedia transcoding, and especially video processing, is a very complex and hardly predictable task, i.e., some of its parameters of work (e.g., the processing time) are difficult to predict. Moreover, the best-effort algorithms are not designed to be controlled during execution in respect to the amount of processed data and time constraints. The idea of using MD to avoid above problems, and thus to allow data independence in MMDBMS, is the only possible solution in our opinion.

As mentioned already, the MD is generated during the non-real-time content analysis of an MO, which is meant to be a separate module. For simplicity, however, our prototype integrates the step of content analysis with the step of the transformation from the source format to the internal storage format. So, in case of video, the encoding to LLV1 in non-real-time phase includes additional content analysis activities, which deliver the required statistical data describing the structure of the lossless bit stream used for scheduling as well as the data required for the encoding process in real-time transcoding.

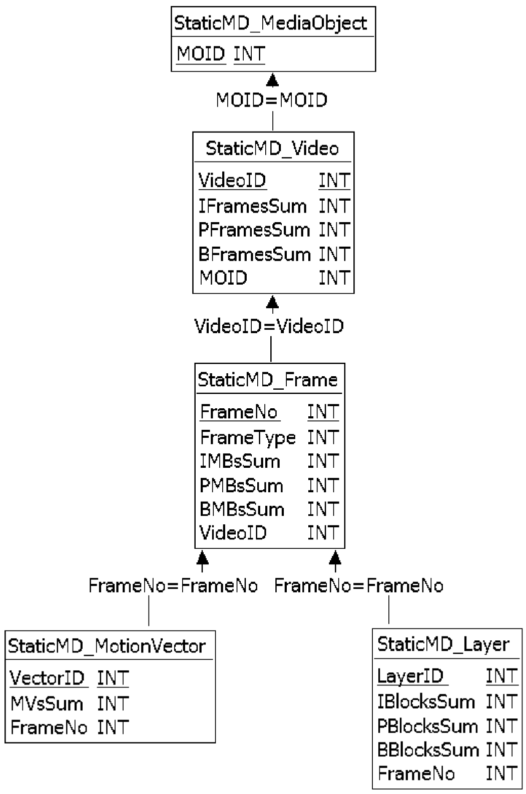
Two main types of MD can be distinguished: static (coarse-granularity data) and continuous (fine-granularity data). The two have different purposes in the media server.

5.1 Static Meta-Data

The *static MD* describe the MO as it is stored and hold information about the structure of the video and audio binary streams. So, static MD keep statistical data allowing accurate prediction of resource allocation for the media transformation in real-time. Thus, they must be available before the real transcoding starts. However, static MD are not required anymore during the process of transcoding, so they may be stored separately from the MO. A noticeable fact is that the size of static MD is very small in comparison with fine-granularity data (continuous MD).

The current definition of static MD of the video type mapped to the relational schema is presented in Figure 2. We assume that an MO is uniquely identifiable and may consist of one or more video or audio streams, so the static MD belonging to the MO are referred by MO identifier. Static MD of the video stream, which are uniquely identified within the MO, include sums of each type of frames. Next, the static MD are calculated per each frame in the video – namely they include a frame number and type, and sums of each macro-block types. The information about sum of different block types is stored in respect to the layer type (in LLV1 there are four layers defined – one base and three enhancement layers (Militzer et al., 2005)). And finally, the sum of different motion vector types is kept for the frame. Nine types of vectors, which could be used in the video, are recognised up to now.

Figure 2 Static MD for the video type



5.2 Continuous Meta-Data

The *continuous MD* are time-dependent (like the video and audio data itself) and are to be stored together with the media bit stream to guarantee real-time delivery of data. The continuous MD are meant for helping the real-time encoding process by feeding it with the information prepared by the content analysis step, which has been placed in the non-real-time part. In other words, they are required to reduce the complexity and unpredictability of the real-time encoding process. Owing to the paper size limitation, we consider only the video-specific MD.

Few kinds of information are stored as continuous MD for video, namely: *frame coding type and bipred value*, *MB width and height, MB mode and priority*, three most significant *coefficient values* and *motion vector(s)*. Especially, fine-granularity information on coefficient values and motion vectors makes the size relatively large, and thus continuous MD must be compressed along with the media bit stream (yielding about 2% of the LLV1 stream size). Since frame coding type, macro-block coding type and motion vectors have been explained in Suchomski et al. (2005), the following explanation covers only the other remaining kinds of continuous MD.

Bipred value. In addition to macro-block type, the bipred value says whether two vectors have been used to predict the block.

Macro-block width and height. This is just a simple information about the number of MB in two directions, horizontal and vertical, respectively. It allows us to calculate the number of MBs in the frame (we do not assume that it is always the same).

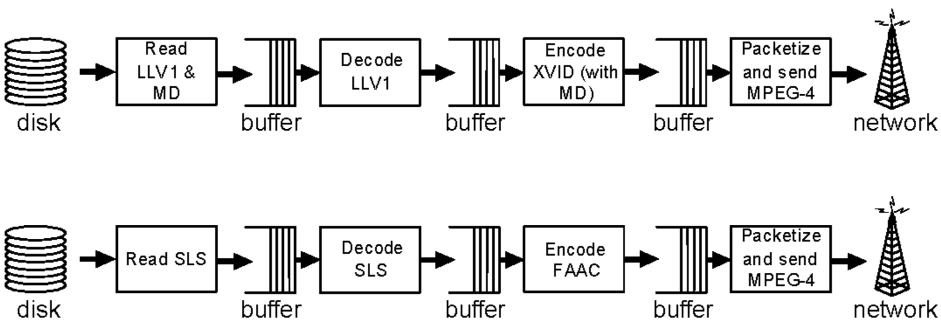
Macro-block priority. If the priority of MB was considered in addition to macro-block type, the processing could be influenced by calculating at first these MBs with the highest priority, which btw. is done in our implementation for the intra blocks (they have the highest priority).

Most significant coefficients. The DC and two AC coefficients are stored in addition but only for the intra-coded blocks. This allows for better processing control by giving possibility of skipping coefficient calculations in case of lack of resources (in such situation we would just calculate not the real but estimated values and still deliver acceptable picture quality).

6 Conversion example and evaluation

We have used the well-known XviD codec and the LLV1 codec as the base of our prototypical implementation for video, and the MPEG-4 SLS reference implementation together with open-source FAAC (compatible with MPEG-4 AAC) for audio. So, four simple steps in our example of video/audio conversion are performed (Figure 3), namely the LLV1/SLS stream together with accompanying MD is read from the hard drive, next it is decoded to raw data and encoded to MPEG-4 SP/AAC and finally it is packetised and sent to the network.

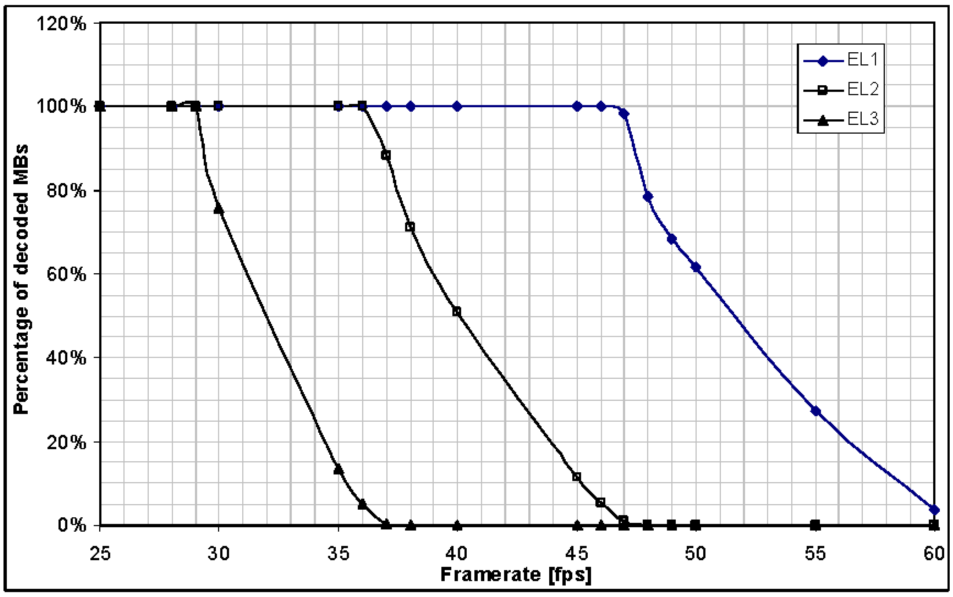
Figure 3 Transcoding from internal formats to MPEG-4 compatible for video (upper graph) and audio (lower graph)



As the RETAVIC framework proposes, we have split the video encoding into two parts: content analysis, which is encapsulated in the LLV1 encoder (in the non-real-time preparation phase), and MD-based encoding using the continuous MD delivered from the outside (read from the hard disk). It has been shown in Suchomski et al. (2005) that the output quality of both ‘Tempete’ and ‘Salesman’ sequences compressed with use of MD (instead of running regular ME) is equal to the output quality when using unmodified XviD regardless of the requested bit-rate. Second, the behaviour of the encoder has become stable and thus more predictable resulting in smaller differences between max and min frame processing time. Thus, we are able to accurately determine resource requirements and adopt stricter buffer techniques for continuous data processing. The reuse of MVs, frame- and MB-types turns out even more beneficial when considering worst-case scenarios, e.g., sequences featuring highly irregular, unpredictable motion (Suchomski et al., 2005). Moreover, the idea of reusing ME-based MD in video transcoding applies also to different bit-rates. It is easily noticeable that the execution times of the encoder using MVs allow for processing much more frames per second than unmodified XviD, and lowering encoder bit-rates yields in a higher speed-up.

Next, we wanted to check support for QoS control, i.e., if the framework designed as proposed will allow for gaining some control over the whole transcoding process. The non-real-time preparation phase allows for reduced video encoder complexity, but also makes LLV1 decoding take up a larger part in the overall video transcoding process. Hence, the scalable LLV1 decoding (Militzer et al., 2005) allows gaining significant control over the overall transcoding process as shown in Figure 4. So, when using only the Base Layer (BL) without enhancement layers (i.e., 0% for all three EL1, EL2, EL3), we have obtained PSNR equal to 32 dB and execution time up to 16 ms per frame (achieving on the graph over 60 fps). When we have additionally selected two quantisation enhancement layers (i.e., 100% of EL1 and EL2, and 0% of EL3), the PSNR value achieved 44 dB and the processing has taken 28 ms per frame (about 36 fps). Moreover, Figure 4 shows that all video qualities between no loss (all enhancement layers decoded, 30 fps) and base quality on PSNR level of 32 dB (all ELs have 0% MBs processed) are achievable.

Figure 4 Percentage of decoded MBs for enhancement layers in LLV1 (Mobile/CIF) with increasing frame rate (see online version for colours)



In respect to audio, the scalability of quality has been evaluated in Suchomski et al. (2006), where the input SLS bitstream was truncated to different sizes and decoded. Furthermore, the test was done for different bit rates of the core layer, namely for cores equal to zero (non-core), 64 kbps and 128 kbps. The resulting audio data were then compared with the reference files using the Objective Difference Grade (Suchomski et al., 2006). In results (due to the paper size limitation, no graphs are included), the scalability of SLS has been proven to be very efficient in respect to quality gain compared with the added enhancement bits and the biggest quality gain is achieved when scaling towards bit rates around 128 kbps. However, the processing scalability of SLS has not met completely our expectations, but still two levels of processing complexity have been possible: using just the core layer or using both, the core and enhancement layers.

All in all the RETAVIC framework, where the MD is used, shows much more stable behaviour of the audio and video transcoding. When considering different quality ranges up to full-content decoding of LLV1 and encoding to XviD by exploiting continuous MD, the quality of data as well as quality of service can be better controlled. When considering audio, the transcoding is controllable in respect to data quality, but in respect to processing can be controlled only to some extent.

7 Conclusions and further work

Our goal has been to develop an architecture allowing for format independence of MOs by applying real-time transformations on demand, thus achieving the data independence in the MMDBMS. The core of RETAVIC real-time transformation framework has been described in detail, i.e., two phases have been explained: preparing MOs for storage including content analysis, and transforming stored data in real-time to the format requested by user. The internal storage formats of media data for application in the MMDBMS have been proposed for both media types, namely the layered lossless format LLV1 for video and MPEG-4 SLS for audio.

Making use of MD, which was described in detail, seems to be a very promising idea for the purpose of multi-format delivery in media servers. As the evaluation showed, we can gain control over the complex and unpredictable algorithms used for audio and video transcoding. Second, the compression effectiveness in respect to the time can be raised, thus allowing to serve more requests than the usual best-effort algorithms. And finally, the QoS can now be controlled with a higher degree owing to the layered storage format and real-time processing.

The continuous as well as static MD sets are not fixed yet, so it is a good point for further investigation and for defining extensions to make any video and audio transcoding schedulable on a real-time operating system. Also, real-time implementation of the mentioned transcoding scenario has to be developed to prove the QoS controllability and feasible schedule ability on real-time operating system – already some work is started in this direction.

References

Bente, N. (2004) *Comparison of Multimedia Servers Available on Nowadays Market – Hardware and Software*, Study Project, Database System Chair, FAU Erlangen-Nuremberg, Erlangen, November, Germany.

Boll, S. (2005) 'Image and video retrieval from a user-centered mobile multimedia perspective', *Proc. 4th International Conference on Image and Video Retrieval (CIVR'05)*, LCNS 3568, July, Singapore, pp.18–27.

Geiger, R. and Yu, R. (2005) MPEG-4 Scalable Lossless Coding, *ISO/IEC JTC1/SC29/WG11 N7707*, October, Nice, FR.

ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC (2005) Advanced Video Coding For Generic Audiovisual Services, *ITU-T, May*.

Kuhn, P. and Suzuki, T. (2001) 'MPEG-7 metadata for video transcoding: motion and difficulty hints', in Yeung, M.M., Li, C-S. and Lienhart, R.W. (Eds.): *Proceedings of the SPIE Conference on Storage and Retrieval for Multimedia Databases*, San Jose, CA, Vol. 4315, p.352, DOI: 10.1117/12.410945 Online Publication Date: 20 May 2003, pp.352–361.

Melton, J. and Eisenberg, A. (2001) 'SQL multimedia and application packages (SQL/MM)', *ACM SIGMOD Record*, December, Vol. 30, No. 4, pp.97–102.

Militzer, M., Suchomski, M. and Meyer-Wegener, K. (2005) 'LLV1 – layered lossless video format supporting multimedia servers during realtime delivery', *Proc. SPIE Vol. 6015. Multimedia Systems and Applications VIII in conjuction to OpticsEast*, October, Boston, MA, pp.436–445.

MPEG-4 Standard 14496-2 (2001) *Information Technology – Coding of Audio Visual Objects – Part 2 Visual*, 2nd ed., ISO/IEC 14496-2.

MPEG-7 Standard 15938-5 (2003) *Information Technology – Multimedia Content Description Interface – Part 5 Multimedia Description Schemes*, ISO/IEC 15938-5 Chapter 8 Media Description Tools, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=34232

MPEG-21 Standard 21000-1 (2004) *Information Technology – Multimedia Framework (MPEG-21) – Part 1 Vision*, Technologies and Strategy, ISO/IEC 21000-1, http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40611

Scherp, A. and Boll, S. (2005) 'Paving the last mile for multi-channel multimedia presentation generation', *Proc. IEEE 11th International Conference on Multimedia Modeling (MMM'05)*, Melbourne, Australia, pp.190–197.

Suchomski, M., Märcz, A. and Meyer-Wegener, K. (2004) 'Multimedia conversion with the focus on continuous media', in van Bommel, P. (Ed.): *Transformation of Knowledge, Information and Data (Chapter XI)*, Information Science Publishing, London, UK, pp.235–256.

Suchomski, M., Militzer, M. and Meyer-Wegener, K. (2005) 'RETAVIC: using meta-data for real-time video encoding in multimedia servers', *Proc. ACM NOSSDAV'05*, Skamania, WA, June, pp.81–86.

Suchomski, M., Meyer-Wegener, K. and Penzkofer, F. (2006) 'Application of MPEG-4 SLS in MMDBMSs – requirements for and evaluation of the format', *Proc. 120th AES Convention*, May, Paris, France, http://www.aes.org/e-lib/browse.cfm?elib=13533.

Sun, H., Chen, X. and Chiang, T. (2005) *Digital Video Transcoding for Transmission and Storage*, CRC Press, Boca Raton, FL, Chapter 11, pp.391–413.

Suzuki, T. and Kuhn, P. (2000) *A Proposal For Segment-Based Transcoding Hints*, ISO/IEC M5847, Noordwijkerhout, Netherlands.

Tourapis, A.M. (2002) 'Enhanced predictive zonal search for single and multiple frame motion estimation', *Proc. SPIE Vol. 4671 – Visual Communications and Image Processing 2002*, pp.1069–1079.

Vetro, A., Sun, H. and Divakaran, A. (2000) *Adaptive Object-based Transcoding using Shape and Motion-based Hints*, ISO/IEC M6088, Geneva, Switzerland.

Vetro, A., Christopoulos, C. and Sun, H. (2003) 'Video transcoding architectures and techniques: an overview', *IEEE Signal Processing Magazine*, March, Vol. 20, No. 2, pp.18–29.

Vetro, A. (2003) 'Transcoding scalable coding and standardized metadata', *International Workshop Very Low Bitrate Video (VLBV) Vol. 2849*, September, Urbana, IL, pp.15–16.

Vetro, A. (2004) 'MPEG-21 digital item adaptation: enabling universal media access', *IEEE Multimedia*, Vol. 11, pp.84–87.

Westerink, P.H., Rajagopalan, R. and Gonzales, C.A. (1999) 'Two-pass MPEG-2 variable bit-rate encoding', *IBM Journal of Research and Development*, Vol. 43, No. 4, p.471.