Real-time Visualisation and Browsing of a Distributed Video Database

Eduard Gonzales, Alun Evans, Sergi Gonzales, Juan Abadia, Josep Blat

Universitat Pompeu Fabra – Fundació Barcelona Media

Tanger 122 - 140

08018 Barcelona, Spain

+34 93 542 20 00

alun.evans@upf.edu josep.blat@upf.edu

ABSTRACT

We present the initial results of a system designed to visualise and browse a distributed database of high definition video. Multiple video clips are taken from the database and streamed as textures within a 3D environment. The rotation, movement, and zoom that this allows enables the clips to be organised, previewed, and selected for further editing. The power of the system lies in the ability to stream multiple clips in real time (including multiple versions of the same clip) and organize the clips according to simple metadata. This latter ability allows a typical user, such as a production director or editor, to rapidly discover relevant or related clips and preview the application of several different post-production filters in real-time, increasing the usability and speed of a typical production workflow.

Categories and Subject Descriptors

I.3.8 [**Computer Graphics**]: Applications

General Terms

Performance, Design, Management.

Keywords

Video, Film, visualization, tool, application.

1. INTRODUCTION

The complexity of audiovisual productions using computer graphics has grown enormously during the last decade, and continues increase with no foreseeable limit. Film and television visual effects and the videogames industry have pushed forward the development of technologies and algorithms to increase the visual quality of productions. The number of people involved has also been increasing and teams are now composed of specialists from diverse disciplines. Artists generate very large sets of assets while developers write the code to handle them. The push for visual quality leads to new generations of graphics programming.

As far as the developer is concerned, every generation makes it necessary to re-write the rendering code: a new or modified algorithm potentially changes the whole rendering workflow. Against this background, the classic Computer Science formula "Programs = Data Structures + Algorithms" is becoming much less useful. One strategy to alleviate the current problems is to enable the reuse of existing workflows across different generations of graphics. This is the approach taken by the iMP FP7 EU project [1], which provides the basis for this work. iMP aims to create architecture, workflow and applications to support the intelligent metadata-driven processing and distribution of digital movies and entertainment programmes

In this work we present a real-time streaming front-end browser for a distributed database of high definition video assets. The application permits navigation through database-stored video sequences using a predefined set of display styles. The database interface is provided by a PostgreSQL database, built on custom hardware that masks a high speed link to several different storage servers (the details of which are not presented here). The front end application can be thought of as similar to Apple's Coverflow browser [2], or the CoolIris image browser[3], with the important extension to be able to deal directly with multiple video streams. Thus main technical developments of our system are both the ability to stream several video clips from the database onto the same screen in real time, use simple metadata to organise them, and possibly apply post-production algorithms to video textures within the GPU pipeline.

2. IMPLEMENTATION

Once a video stream is received after a call to the database, the application reads an XML scene definition that defines both what information should be displayed and what style should be used to display it. The video is then applied as a texture overlay for a flat rectangular surface, drawn in 3D space using our custom 3D engine. The shape of the video textured surface (VTS) is set according to the aspect ratio of the corresponding video clip, and the VTS texture is updated according to the clip's framerate. Multiple VTSs can be displayed on the screen simultaneously and arranged in a variety of ways (see Section 3 below). The user is able to manipulate the VTS within the 3D environment, with the system applying standard translation, rotation and scaling transformation to each VTS, while still maintaining real-time video texture update. The image resolution of the video test sequences presented here is 624 by 352 and the framerate is 23.97.

Maintaining multiple video streams in memory is a difficult task that requires careful management of the flow of data. To facilitate this, culling algorithms are used to avoid updating the video on

VTSs that are not visible in screen, thus reducing the video memory bandwith. In order to display several videos simultaneously in realtime, we have stored all the images in DDS (DirectDraw Surface) using DXT1 compression. This compression format is hardware accelerated in most of current consumer graphics cards. With this configuration, we are able to display between 8 to 12 video sequences at the same time without visible performance problems.

The iMP database is a PostgreSQL database. Instead of using a proprietary PostgreSQL middleware to connect our application with the database, we have decided to use ODBC in order to change the database without rewriting the code. After evaluating several middlewares, we chose the open source middleware libodbc++ [4] due to its clean interface. In order to not to block the application while loading the video sequences, the prototype makes use of multithreading programming. We have two working threads dedicated solely to their individual tasks. One thread is responsible for loading the video sequences in main memory and the other one is responsible to render the scene.

The prototype application presented supports two interesting capabilities that should form the basis for future work. The first is the possibility to organise clips via associated metadata. As a proof-of-concept we tagged a set of video sequences with actor metadata information and created an XML scene definition example that lets the user filter the video sequences by choosing the actor name. A further interesting implementational detail is the ability to support the viewing of several versions of the same clip, while applying post-production algorithms to the VTS texture within the GPU shader pipeline. This allows the user to see a preview of application of several different post-production filters simultaneously to a video clip.

3. PRESENTATION

Currently there are four display styles, examples of which are shown in Figures 1-3:

- Classic: shows all the video sequences ordered by row or column (Figure 1)
- Detail: zooms on the central video clip in the sequence (Figure 2)
- Coverflow: shows the video sequences by user-controlled animated scrolling (Figure 3)
- Box: shows the video sequences wrapped in a 3D box

4. FUTURE WORK

This application forms a firm basis for the integration of several technologies. The immediate work will be to focus on adding a more formal and structured metadata framework [5], enabling large databases to be transversed and viewed simply. Further work will allow users to have more control over the appliction of post-production filters applied via the GPU pipeline, thus laying the foundations of a “Virtual Film Factory” which is the goal of the iMP project. During this development process we will initiate a comprehensive evaluation plan, allowing professionals to use and test the integrated tools, providing invaluable feedback for further development.

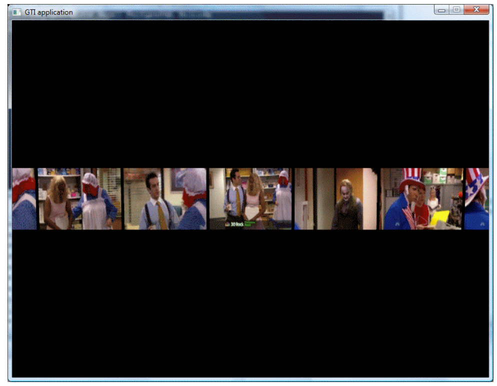

Figure 1: Classic view (row)

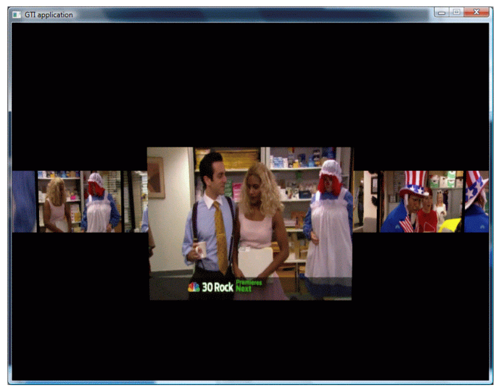

Figure 2: Detail view zoomed on centre clip

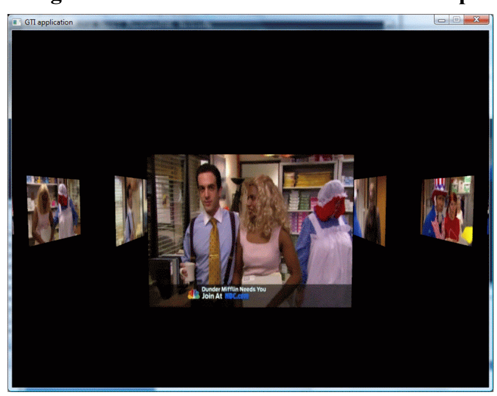

Figure 3: Coverflow view

5. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the European Commission FP7 STREP – IMP [1].

6. REFERENCES

[1] http://imp-project.eu/

[2] http://www.apple.com/itunes/

[3] http://www.cooliris.com/

[4] http://libodbcxx.sourceforge.net/

[5] Hausenblaus, M., Troncy, R., Raimond Y. and Burger, T. 2009. Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments. LDOW 2009, Madrid.