REPRESENTATION, CODING AND INTERACTIVE RENDERING OF HIGH-RESOLUTION PANORAMIC IMAGES AND VIDEO USING MPEG-4

S. Heymann, A. Smolic, K. Mueller, Y. Guo, J. Rurainsky, P. Eisert, T. Wiegand

Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut
Image Processing Department
Einsteinufer 37, 10587 Berlin, Germany

KEY WORDS: Mosaic, Panorama, MPEG-4 Scene Rendering, 3D Scene Representation, Coding

ABSTRACT:

In this paper we present an MPEG-4 compliant system for viewing 360° high-resolution spherical panoramic images and videos. The system allows user navigation within an MPEG-4 3D scene description. Here, two different scene geometries were implemented: One consisting of a spherical geometry around the viewpoint and the other having a cylindrical geometry with associated texture patches each. The textures are either real-world static high-resolution scene images or movie textures This scene dynamically deals with problems like bandwidth and video sizes to provide a real-time viewing experience by dividing the texture into a number of tiles. Thus, only the visible parts of the scene can be rendered. Furthermore, visibility sensors are associated to the texture tiles such that pre-fetching of texture tiles next the visible once is enabled during scene navigation. By varying the tile size and visibility sensor size, experiments for real-time capability of the rendering environment were performed. Additionally, a combination of head tracking device and head mounted display was investigated for a more comfortable scene navigation.

1. INTRODUCTION

Classic Television and Video-Entertainment Systems present prerecorded scenes and movies to the viewer with very limited ways of interaction and with a fixed viewing angle. One step to increase the degree of user-interaction was to introduce different channels for different viewpoints like in common pay-TV broadcasts of sport events. However the interaction with the content is limited to switching between the provided fixed viewpoints. Another approach to overcome such limitations is realized in systems like Quicktime VR [1] where the user can change the viewing direction and zoom in and out of the presented panoramic image. This approach is suitable for static images but has limitations concerning time variant images or video. The Immersive Media Company [8] has then introduced a way to capture and watch panoramic images in a real-time player, which was successfully taken to a commercial level.

The goal of this contribution is to present an efficient MPEG-4 compliant system for representation, coding and interactive rendering of high-resolution panoramic images and videos. We will therefore discuss the problems concerning the issues of bandwidth limitation of common hardware and very large images and videos, which are necessary to provide high image quality in the finally rendered scene. Various publications have shown that omni-directional video applications can be created with existing MPEG-4 components [2], [3], [5], [7]. In this contribution we present a complete system for interactive viewing of omni-directional video based scenes that can be rendered on common MPEG-4 3D Players, e.g. [6]. The scene consists of 3D meshes with associated video textures. We also added head-tracker support to the MPEG-4 3D Player to enable interaction with the system in the most natural way, i.e. by simply turning the head and thus the equipped head-mounted display.

The paper is structured as follows: Section 2 gives an overview of the mosaic construction mainly for the spherical case as the more challenging geometric approach and the integration of interactive scene elements. Standard compatibility with MPEG-4 is shown in Section 3 and Experiments on the real-time capability of the system is presented in Section 4. Finally, conclusions are drawn in the final Section 5.

2. INTERACTIVE PANORAMIC VIDEOS AND IMAGES

Panoramic images and videos are a 360° representation of a certain scene in contrast to an about 60° field of view that is naturally seen by human vision system. As illustrated in Fig. 1, the resolution of the panoramic image source has to be a multiple of common video or image resolution to provide a good quality of rendering as stated within the MPEG-4 3DAV group [4], [5].

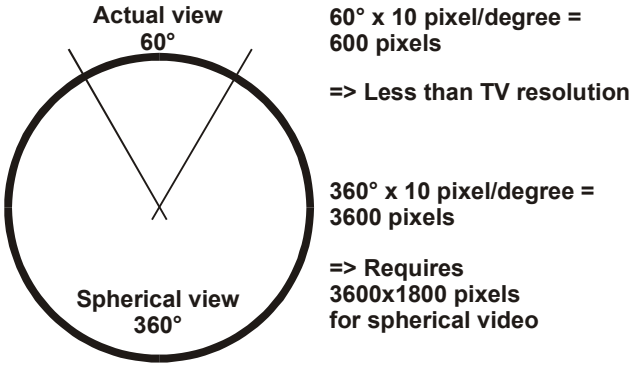

Fig. 1: Requirement for very high-resolution video.

In the example above, the resolution of the actual 60°-view is 600 pixels, which is not that high for a single image. However, the whole 360°-panorama requires a total of 3600 pixels in horizontal direction making it a high-resolution video signal.
Depending on the chosen rendering device, this total resolution is only needed occasionally. If the panoramic scene is not displayed on a dome type of display only a portion of the overall signal needs to be presented at a time. We therefore developed a concept of subdividing the omni-directional video into smaller patches (tiles) as illustrated in Fig. 2.

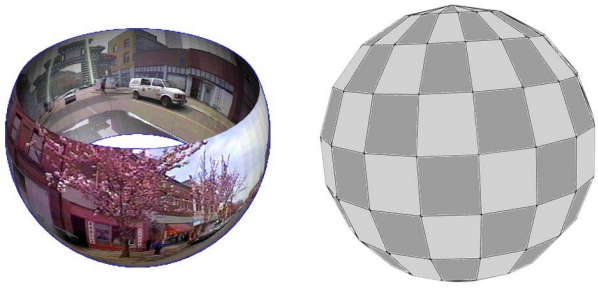

Fig. 2: Division of a sphere into patches.

The benefit of texture tiling is that only parts of the panoramic scene need to be processed at once. Thus, all the tiles are encoded separately [6] to allow independent decoding and rendering. The mapping between texture patches and geometry is similar to that of geodesic panoramic images onto sphere-meshes, as shown in Fig. 3.

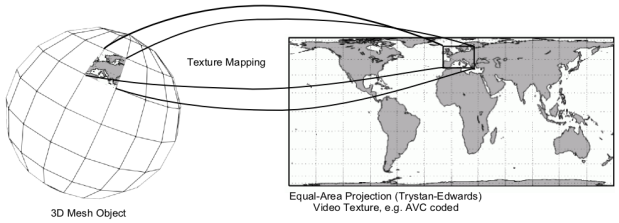

Fig. 3: Mapping of geodesic panoramic images onto sphere-meshes.

Since a flat texture is mapped onto a sphere, texture patches are distorted differently depending on the latitude of the associated geometry surface area. Areas near the poles exhibit the largest distortion, while equatorial patches are hardly deformed. This results in a finally rendered texture resolution, which is highest in equatorial areas. In scenarios, where ad hoc knowledge about the scene geometry is available, such conditions may be exploited for creating different resolution areas in the original texture data. For the cylindrical geometry case of course, different resolution areas do not occur.
Finally, a visibility sensor is assigned to each of the patches as shown in Fig. 4.

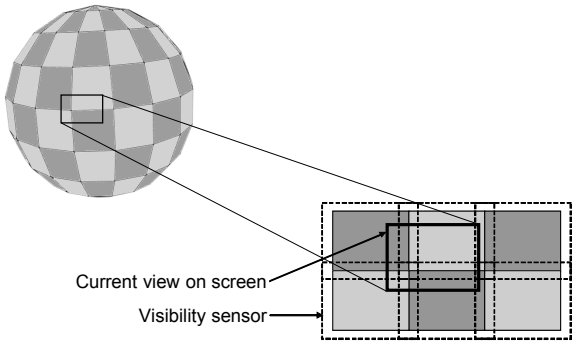

Fig. 4: Video-patches and visibility sensors.

The corresponding video or texture information is only decoded and rendered if the visibility sensor coincides with any part of the visible scene in the rendering window. By over sizing the visibility sensors, a pre-fetching mechanism is implemented that ensures smooth navigation. Texture tiles adjacent to the actually displayed ones are thus already loaded and can be rendered immediately if the navigation direction changes continuously. Thus, the size of the texture patches, as well as the overlapping percentage influence the real-time behavior of the scene. Note, that for abrupt changes like switching between predefined viewpoints the visible scene has to be entirely reloaded into video memory.

2.1 Interactive Scene Elements

Besides the already mentioned interactive system for navigation by view field analysis using visibility sensors, the MPEG-4 technology provides many new features compared to conventional 360° panoramas. Video objects, dynamic 3D computer models [9], [10], or spatial audio as illustrated in Fig. 5 can be embedded in order to vitalize the scene.

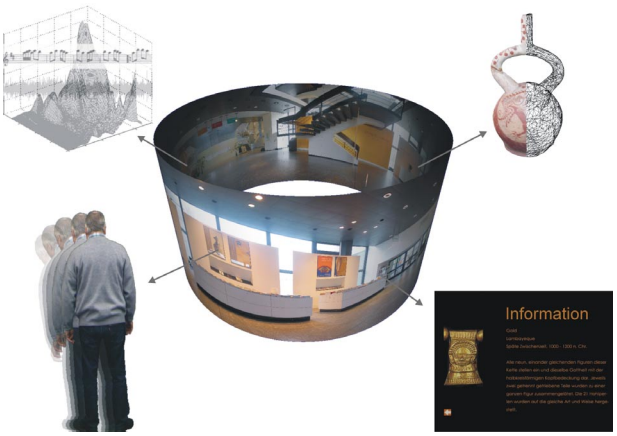

Fig. 5: Scene elements of our MPEG-4 player. Besides the panorama, dynamic video objects, interactive buttons, 3D models or spatial sound can be added to the environment.

Pressing interactive buttons gives additional information about objects or modifies the current location. Thus, large high quality environments can be created that enable the user to immerse into the virtual world.
The possible degrees of freedom for the navigation through a 3D reconstructed real world are specified during the image acquisition step. Rotating the capturing device around the image plane allows the navigation by rotation and zoom. Such viewing restrictions, like rotation and zoom can somewhat be relaxed by allowing to jump between different panoramas as shown in Fig. 7. However, for many applications this is sufficient and panoramic views can be found more and more often on web sites creating virtual tours for city exploration, tourism, sightseeing, and e-commerce.

3. MPEG-4 IMPLEMENTATION

In the first user scenario, we developed a BIFS scene for spherical video as described above and combined it with videos from the 3DAV test set (provided by Immersive Media Co.). The resulting mp4 files can be viewed with any MPEG-4 3D Player such as Fraunhofer HHI's implementation [6]. The user can freely navigate the scene by choosing arbitrary rotation and zoom. This can be done with mouse interaction to get an appropriate view on the screen, as well known from many static panorama applications (such as Quicktime VR).
We also combined the 3D Player with a head-mounted display and tracker as illustrated in Fig. 6.

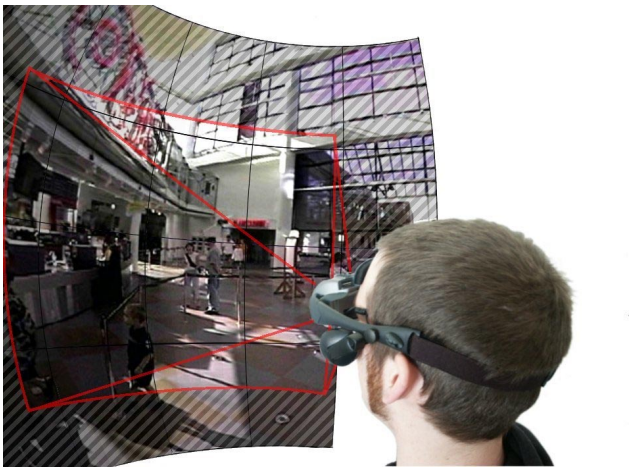

Fig. 6: Head-mounted display navigation: The spherical rectangle is currently visible, while the shaded outside area is pre-fetched.

The system evaluates the head rotation and renders appropriate views onto the display. This creates the immersive impression of being part of the scene. The viewpoint for this architecture is arbitrary in direction, but fixed in position that is identically to the center of the projection sphere. Furthermore, the scene is rendered onto the head-mounted display which together with the real-time tracker input provides the user with an impression of being within a dome-like projection architecture where each viewing direction results in the associated proper movie texture representation Together with the used MPEG-4 renderer, it runs reliably in real-time.
For the second user scenario, a cylindrical panorama was used together with high-resolution still textures and a number of further MPEG-4 scene elements. Since MPEG-4 BIFS provides the full functionality of common 3D scene description languages, like VRML, and additionally offers compression and transmission of 3D scene content, all the described interactive elements could be implemented in a standardized way to build a guided tour of a number of interactively connected cylindrical panoramas.

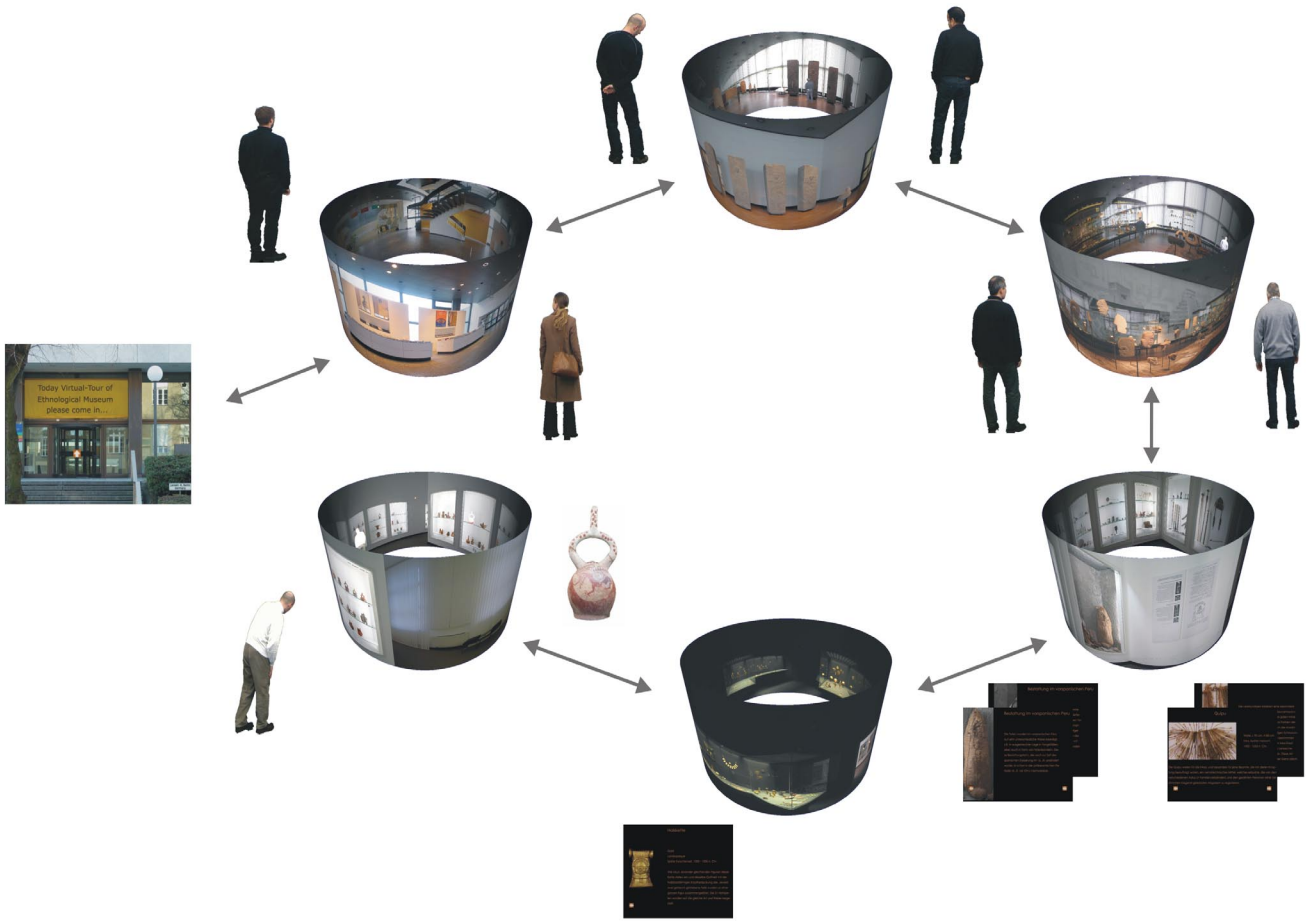

Fig. 7: Multiple panoramas from the Ethnological Museum in Berlin. Interactive scene elements allow the user to jump between the rooms. Dynamic objects are added to vitalize the scene.

Currently, there is ongoing work in MPEG-4 to investigated efficient coding tools for 3D geometry and multi-view video coding from which the introduced scene elements can also benefit.

4. EXPERIMENTS

One of the most important issues in the experiments was to optimize the tradeoff between tile size and number of sphere segments. With oversized segments the pre-fetching mechanism becomes ineffective, since a new segment is always loaded even if a very small part is visible, i.e. as soon as the visibility sensor is activated. The same applies to disappearing segments, which are also processed all the time until they have disappeared completely. In such cases, the produced amount of invisible data can be even higher than the displayed data.

On the other hand using a large number of small sized segments the visibility tests of the system become too expensive and the synchronization of the used video patches cannot be achieved in time, especially in cases of fast changes in viewing direction.

The used image information from a 3600x1800 pixel source image, which is displayed at a time, is about 600x450 assuming a 60° field-of-view (see Fig. 1) and has an aspect ratio of 4:3. Therefore the data to be displayed has an uncompressed size of approximately 1 Mbyte. If the whole image had to be loaded into video RAM, the image data required about 25 Mbytes for each single frame. Thus, an overhead of 24 Mbytes for each frame would be computed for this example setup to reside in video memory. After navigation change, most of this data becomes obsolete, as it is outside the actual navigation path and therefore never gets displayed.

With our pre-fetching mechanism, a video tile is just loaded when it is about to appear in the field-of-view which drastically reduces the mentioned overhead. Table 1 shows a comparison of the produced overheads using different tile sizes and sphere subdivisions.

	tile size (degrees at equator)	avg. over-head	number of video-tiles
1.	5.625	0.08 MB	2048
2.	11.25	0.23 MB	512
3.	22.5	0.68 MB	128
4.	45	2.0 MB	32
5.	90	5.7 MB	8

Table 1: Relation between tile sizes and data overhead with visible part of the panoramic image of about 1 MB.

We decided to take a sphere divided into 128 segments as seen in Table 1 row 3. With one visibility sensor for each segment we also have 128 of these sensors, which causes an acceptable number of operations to be performed to get the visibility information. Using 32 or 8 segments on one hand produce a large video overhead, which results in synchronization problems for the video-patches, using 2048 small segments on the other hand make the visibility calculations to expensive. The latter case causes a delay of hidden-to-visible swaps, which results in video-patches popping up at the sides of the screen in the case of faster head rotations.

5. CONCLUSIONS AND FUTURE WORK

We have demonstrated that omni-directional video applications can be created efficiently using existing MPEG-4 technology by combining a spherical or cylindrical geometry and tiled omni-directional high-resolution video streams or still image textures. Furthermore, a visibility sensor was attached to each video tile to provide a pre-fetching mechanism for efficient video memory usage and guarantee real-time rendering. The experiments showed, how to select the overlapping visibility sensor area in comparison to the size of the video tiles to provide real-time rendering.

The assumption of random access, which is important to ensure smooth real-time rendering, can be resolved using INTRA-only coding. Very efficient ways to do this are JPEG2000 or AVC INTRA-only. Within a streaming environment, where high-resolution texture information is transmitted online, only the video streams of active textures could be transmitted. For future work, the actual viewing direction could than be used at the server to request only visible video tiles plus the adjacent tiles for pre-fetching. If today's video coding technology is used with predictive P and B frames, full random access is not possible, since intra frame information is required for decoding. Here a tradeoff between I frame period, multi frame memory and active pre-fetching area would be required to allow decoded images to be rendered in time.

REFERENCES

[1] S. E. Chen, "Quicktime VR – An Image-Based Approach to Virtual Environment Navigation" *Computer Graphics*, Vol.29, pp.29-38, 1995.

[2] P. Eisert, Y. Guo, A. Riechers and J. Rurainsky, "High-Resolution Interactive Panoramas with MPEG-4", *in Proc. of the 9th Workshop for Vision Modeling and Visualization*, Stanford (California), USA, November 2004

[3] C. Grünheit, A. Smolic, and T. Wiegand, "Efficient Representation and Interactive Streaming of High-Resolution Panoramic Views," *Proc. ICIP'02, IEEE International Conference on Image Processing*, Rochester, NY, USA, September 22.-25. 2002.

[4] ISO/IEC JTC1/SC29/WG11, "Applications and Requirements for 3DAV", Doc. N5877, Trondheim, Norway, July 2003.

[5] ISO/IEC JTC1/SC29/WG11, "Report on 3DAV Exploration", Doc. N5878, Trondheim, Norway, July 2003.

[6] A. Smolic, Y. Guo, J. Guether and T. Selinger, "Demonstration of Streaming of MPEG-4 3-D Scenes with Live Video," *ISO/IEC JTC1/SC29/WG11*, Doc. M7811, Pattaya, Thailand, December 2001.

[7] A. Smolic, D. McCutchen, "3DAV Exploration of Video-Based Rendering Technology in MPEG", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, no. 9, pp. 348-356, 2004.

[8] www.immersivemedia.com

[9] I. Feldmann, P. Eisert, and P. Kauff, „Extension of epipolar image analysis to circular camera movements", *In Proc. International Conference on Image Processing (ICIP)*, pp. 697-700, Barcelona, Spain, Sep. 2004.

[10] P. Eisert, "3-D geometry enhancement by contour optimization in turntable sequences, *In Proc. International Conference on Image Processing (ICIP),* pp. 3021-3024, Singapore, Oct. 2004.