# H.264 BASED CODING OF OMNIDIRECTIONAL VIDEO

Ingo Bauermann, Matthias Mielke and Eckehard Steinbach
*Technische Universität München, Institute of Communication Networks, Media Technology Group*

**Abstract:** Omnidirectional video is an adequate image-based scene representation data format for interactive walkthroughs or surround viewing of a scene over the Internet. While efficient compression is a key issue in this context, the properties of omnidirectional video are not in line with the assumptions that are made about video sequences by modern compression standards like MPEG-4 AVC/H.264. We introduce a preprocessing approach which transforms omnidirectional video into a sequence of panoramic images before encoding. Using a state of the art MPEG-4 AVC/H.264 video coder, our approach performs up to 2dB better for low bit rates compared to regular encoding of omnidirectional video.

**Key words:** omnidirectional video, video compression

## 1. INTRODUCTION

Omnidirectional video systems are capable of capturing a 360-degree horizontal field of view at one time. This wide view characteristic makes them suitable for applications like interactive walkthroughs and surveillance (e.g. [1,2]). Figure 1 shows one frame of an omnidirectional video sequence and a perspective view generated from one portion of the frame.

This kind of dataset allows pan, tilt, zoom, and even translational movement along the specific path the sequence was captured on. While the main advantage of omnidirectional video is the wide field of view, the specific geometric acquisition properties that lead to it contradict some assumptions common video compression schemes are based on. To address this we propose a coding scheme based on preprocessing of the captured omnidirectional video sequence and subsequent encoding using the state-of-the-art video codec MPEG-4 AVC/H.264. Figure 2 shows the processing pipelines for direct encoding of omnidirectional video (DEOV) and coding of preprocessed omnidirectional video (CPOV) investigated in this paper.

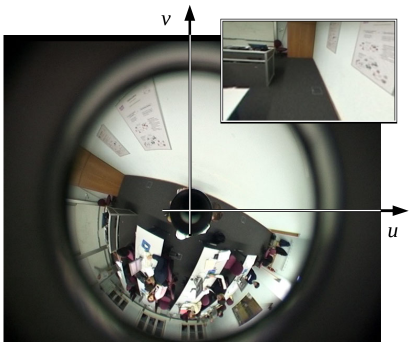


*Figure 1.* One frame (720x576) from the "classroom" test sequence. The upper right shows a perspective mapping from the original ominidirectional image.

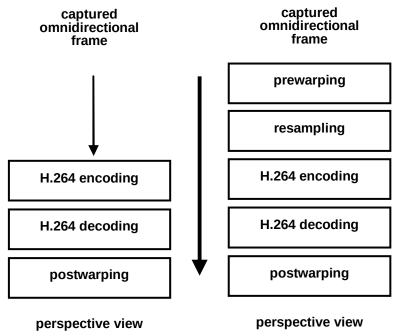


*Figure 2.* Processing pipelines (left) for direct encoding of omnidirectional video sequences (DEOV) and (right) for coding of preprocessed omnidirectional video (CPOV).

In the following some issues for the DEOV approach are given that motivate our CPOV approach. The number of reference pixels in the omnidirectional view used to interpolate a pixel in the perspective view is a function of image coordinates (u,v) in Figure 1. Depending on the shape of the mirror and the lenses used for capturing the omnidirectional video, the introduced distortion in the final view is inhomogeneous. For DEOV blocking-artifacts caused by rate-constrained compression of the omnidirectional video result in jagged vertical edges. The distribution and shape of blocking-artifacts change for different viewing directions (see Figure 8). Motion estimation and compensation performed in standard video coders assume translational motion of small image blocks from one video frame to another. For omnidirectional video this assumption does not hold. Motion of the camera or in the scene results in a change of orientation and scaling of corresponding image blocks in the omnidirectional image. In addition, reconstructing perspective views directly from captured omnidirectional frames is computationally intense. Projection properties are often non-linear and depend on the viewing direction.

To adapt to some extent to these properties, the proposed coding scheme CPOV performs prewarping and subsampling steps on the captured sequence before encoding. The resulting panoramic sequence is resampled in vertical and horizontal direction before it is fed to the video encoder.

The catadioptric imaging system used in this work uses a hyperbolic mirror and lens system from "Remote Reality" [3]. For the omnidirectional video sequences captured with this device a computationally complex non-linear projection is needed to generate perspective views.

The remainder of the paper is structured as follows. In Chapter 2 we describe the prewarping and subsampling scheme for the preprocessing step. Chapter 3 discusses the impact on the coding efficiency and rendering complexity of our approach. In Chapter 4 experimental results are presented. Chapter 5 gives concluding remarks.

## 2. PREPROCESSING

The calibration of camera and mirror parameters is performed using the algorithm described in [4] and [5]. The pre-processing step consists of two main parts. First a panoramic image is generated from the captured omnidirectional frame using bilinear interpolation. Figure 3 shows a panoramic image created from the “classroom sequence”. The mapping from the omnidirectional image plane to the panorama is illustrated in Figure 5. Pixels of size $\delta A$ are mapped to an area $\delta P$ on the panorama plane via the mirror surface. The area ratio between the two representations is given by

$$\theta = \frac{\delta P}{\delta A} = \begin{pmatrix} \frac{u(s,t)}{dt} \\ \frac{v(s,t)}{dt} \end{pmatrix} \times \begin{pmatrix} \frac{u(s,t)}{ds} \\ \frac{v(s,t)}{ds} \end{pmatrix}.$$

Figure 4 shows $\theta$ as a function of *t* for a panorama size of 2500x343 pixels.


*Figure 3.* Panoramic image obtained after prewarping the omnidirectional image.

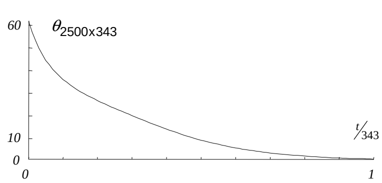


*Figure 4.* Number of output pixels per input pixel $\Theta_{2500x343}$ for a given panorama size of 2500x343.

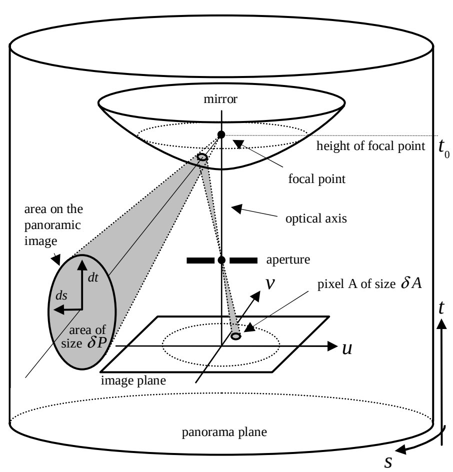


*Figure 5.* Projection from the image plane to the panorama plane.

The resampling step scales the image dimensions independently in horizontal and vertical direction. Let $s_s$ and $s_t$ be the scaling factor for the $s$ dimension and the $t$ dimension, respectively. Then

$$\theta' = \frac{\delta P}{\delta A} \cdot s_s \cdot s_t$$

is the output pixel per input pixel ratio after the pre-processing step. Parameters $s_s$ and $s_t$ are chosen by optimizing the rate distortion trade off for the rendered view. The resulting sequence is fed to the video encoder.

## 3. IMPACT OF PREPROCESSING ON THE CODING STEP

The preprojection and resampling steps in CPOV allow us to adjust spatial resolution and quality in different areas of the rendered view reconstructed from the panoramic image without changing the quantization policy of the used standard video coder. The motivation for this is that the rate-distortion optimization algorithms of the video coder can not take into account that warping steps have to be performed after decoding to generate perspective views. Spatial adaptation of the quantization parameters would solve this problem. This would mean that the encoder has to be modified in such a way that quantization parameters are adjusted during the encoding process according to the current image coordinates. Though no modification of the decoder is needed in this case the client side viewing application has to be aware of the intrinsic parameters of the specific capturing device. In CPOV a general panoramic representation for omnidirectional video is used. This representation is independent of the actual capturing device.

In the rendered view blocking boundaries have constant shape due to the panoramic representation in CPOV. For DEOV the block size increases from top to bottom. For the same reasons motion compensation performs better on the preprojected frame as the motion model used in most standard coders assumes translational motion of image blocks from one frame to the next. The preprojection undistorts the trajectories of linearly moving objects to some extent and compensates for scaling effects.

In CPOV rendering perspective views from the compressed omnidirectional video becomes less complex as the mapping from the image plane $(x, y)$ to the normalized panoramic image coordinates $(s, t)$ given the height $t_0$ of the focal point in the panoramic image and the focal length of the virtual camera $f_v$ simplifies to:

$$s = \frac{\operatorname{atan}\left(\frac{x}{f_v}\right)}{2\pi} \; ; \; t = \frac{y \cdot \cos(2\pi s)}{2\pi f_v} + t_0 \; ;$$

For CPOV the mapping from the panoramic image coordinates $(s,t)$ to the omnidirectional image coordinates $(u,v)$ and mirror calibration is performed during the preprocessing step:

$$u = (1-t)\cdot c_{i,u} + t\cdot c_{o,u} + \sin(2\pi s)\cdot\left((1-t)\cdot P_i^5(t) + t\cdot P_o^5(t)\right)\cdot\left((1-t)\cdot\frac{r_{i,u}}{r_{i,v}} + t\cdot\frac{r_{o,u}}{r_{o,v}}\right)$$

$$v = (1-t)\cdot c_{i,v} + t\cdot c_{o,v} + \cos(2\pi s)\cdot\left((1-t)\cdot P_i^5(t) + t\cdot P_o^5(t)\right)$$

Here $c_{i,u}$, $c_{i,v}$, $c_{o,u}$, $c_{o,v}$ and $r_{i,u}$, $r_{i,v}$, $r_{o,u}$, $r_{o,v}$ denote the centers and radii of the inner and outer calibration circles (see [5]). $P_i^5$, $P_o^5$ denote $5^{th}$ order polynoms calibrating the radial distortion. From the applications point of view the representation proposed here is independent from the geometrical assembly and calibration of the capturing device.

## 4. RESULTS

To evaluate the performance of CPOV a viewer has been developed capable of rendering identical views from the originally captured omnidirectional frame and from reconstructed frames encoded using CPOV and DEOV. The window size is 640x480 pixels. The test sequence is "classroom" a 40 frame video captured with an omnidirectional camera from Remote Reality [4]. Figure 6 shows the rate distortion plot for both CPOV and DEOV. Rendered views from the reconstructed frames are compared to the view rendered from the uncompressed original omnidirectional frame. The rate is measured in bit per rendered pixel where the file sizes of the compressed representations are weighted by the number of rendered pixels in the field of view of the virtual camera. For low rates the gain of CPOV over DEOV is up to 2dB. Figure 7 shows a similar rate distortion plot for the lower third of the rendered view. Up to 3dB better PSNR is achieved using CPOV. The performance of the viewer was tested using the video for windows interface and an XviD codec version 1.0.0 RC2. The frame rate for navigating frame by frame was about 30% faster for CPOV.

## 5. CONCLUDING REMARKS

In this paper a preprocessing technique was presented for encoding omnidirectional video sequences efficiently (CPOV). The pre-processed omnidirectional video sequences are encoded using the state-of-the-art video coder MPEG-4 AVC/H.264. By prewarping and resampling the captured frame to a panoramic image an adaptation to the assumptions standard video coders are based on was investigated. A better tradeoff between quality in the upper and lower areas of the rendered view has been achieved. For relatively low rates the approach described here performs even better than encoding the original frame directly in the rate distortion sense. Rendering is simplified and the client is independent from geometrical and optical parameters of the capturing device.

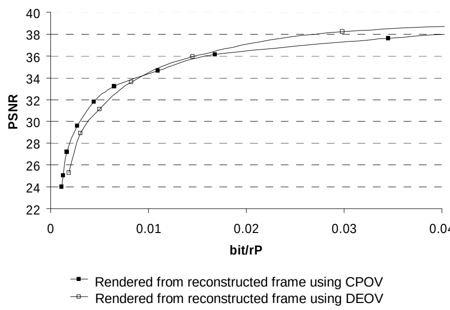


*Figure 6.* Rate-distortion plot for CPOV and DEOV. At rates lower than about 0.01 bit/rendered pixel, CPOV outperforms DEOV.

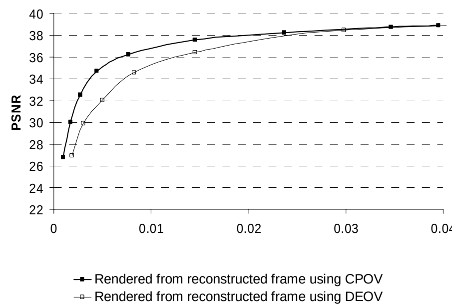


*Figure 7.* Rate-distortion plot for CPOV and DEOV for the lower third of the rendered frame. CPOV outperforms DEOV at all rates as loss due to quantization is minimized in this area.

(a) (b) (c)

*Figure 8.* Views from the lower region generated from (a) the original omnidirectional image (b) the reconstructed panoramic image using CPOV and (c) the reconstructed omnidirectional image using DEOV (same file size as in (b)). In (c) some edges are distorted compared to (b).

## 6. REFERENCES

1. S. E. Chen., "QuickTime VR – An Image Based Approach to Virtual Environment Navigation," Proc. of SIGGRAPH 95, pp. 29-38, Aug. 1995.
2. Y. Onoe, N. Yokoya, K. Yamazawa, and H. Takemura, "Visual surveillance and monitoring system using an omnidirectional video camera," Proc. of International Conference on Pattern Recognition, ICPR'98, pp.588-592, Sept. 1998.
3. http://www.remotereality.com
4. http://www.vision.caltech.edu/bouguetj/calib_doc/
5. J. Fabrizio, J.-P. Tarel, and R. Benosman, "Calibration of Panoramic Catadioptric Sensors Made Easier," Proc. of IEEE Workshop on Omnidirectional Vision, Omnivis'02, pp. 45-52, Copenhagen, Denmark, June 2002.