H.264/AVC-COMPATIBLE CODING OF DYNAMIC LIGHT FIELDS USING TRANSPOSED PICTURE ORDERING

Ulrich Fecker and André Kaup

University of Erlangen-Nuremberg
Chair of Multimedia Communications and Signal Processing
Cauerstraße 7, 91058 Erlangen, Germany
phone: + (49) 9131 85-27108, fax: + (49) 9131 85-28849
email: {fecker, kaup}@lnt.de
web: www.lnt.de/lms

ABSTRACT

A dynamic light field or a multi-view video sequence requires capturing an object or a scene with multiple cameras. This allows viewing the scene from arbitrary viewpoints without the need for geometric information. As this leads to very high data rates, efficient compression is necessary. In this paper, we present an approach to exploiting the spatial correlation between the different camera views in addition to the temporal correlation within each view. A simple resorting scheme is introduced, which allows the usage of an off-the-shelf video coder to compress multi-view video data. Coding results are shown using the H.264/AVC video coding standard. It is discussed how the coding efficiency depends on the frame rate and the camera distance of the sequence.

1. INTRODUCTION

Light fields are an approach to entirely capture the visual information of a three-dimensional object or scene. The intention is to reproduce photorealistic images of the scene for any desired viewpoint and for any viewing angle. In contrast to classical geometry-based approaches, where information such as the geometry and surface characteristics of the object is used to render views from a desired viewpoint, light fields do not need this kind of object description. Instead, a large number of images is taken by multiple cameras from different positions. From these images, intermediate views can be interpolated for viewpoints not coinciding with the original camera positions [1, 3].

A light field can be represented by the seven-dimensional *plenoptic function* $(x,y,z, \ , \ , \ ,t)$, which is the light intensity depending on the viewpoint x,y,z, the viewing angle $\ , \ ,$ the wavelength $\ $ and the time t (see Figure 1). The plenoptic function fully describes the radiance in the space around the desired object or scene. However, this function is far too general and too complex to be dealt with. Therefore, simplifications need to be introduced: Usually, the parameter $\ $ is eliminated by introducing three colour channels, e. g. red, green and blue: $_{R,G,B}(x,y,z, \ , \ , t)$. Furthermore, the function is in most cases not sampled throughout the entire space, but only on a surface surrounding the scene [8]. This configuration is then rather well applicable to a setup where the light emanating from a scene is captured by several cameras surrounding it.

If the recorded object or scene does not change over the time, the parameter t can also be eliminated, and the light field is called a *static light field.* In this paper, the case of *dynamic light fields* is considered, where the light emitted by the object or scene varies over the time. Dynamic light fields could find applications in medicine, where e. g. a surgeon could spatially examine a beating heart, or in a virtual webshop, where customers could view changing three-dimensional objects with complex surfaces from any desired viewpoint.

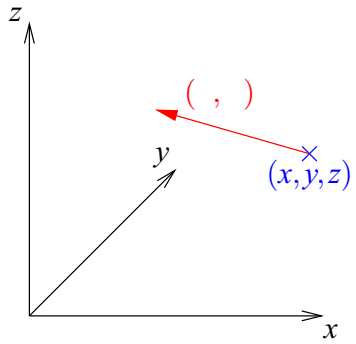

Figure 1: The Plenoptic Function

Another application, which has attracted increasing interest from the industry as well as from research institutes, is *three-dimensional television (3D TV)* or *free viewpoint television (FTV)*. Such systems have e. g. been presented in [7], where the 3D experience is based on an array of multiple projectors, and in [6].

One problem when dealing with video streams from multiple cameras simultaneously is the enormous amount of data. Therefore, efficient compression is needed to realise a practical system. For static light fields, where large numbers of still images need to be stored and transmitted, compression techniques were developed in [4]. In the case of dynamic light fields, when moving pictures are recorded from each camera, the amount of data is even higher. In this paper, an approach is presented which enables the usage of a standard video coder for multi-view video data. The performance is compared to simulcast, where the video stream from each camera is coded separately. Furthermore, the dependency of the results from the camera distance and the frame rate of the recorded sequences is analysed.

2. TEST SEQUENCES

For the coding experiments, different classes of multi-view test datasets have been used, which were provided for the

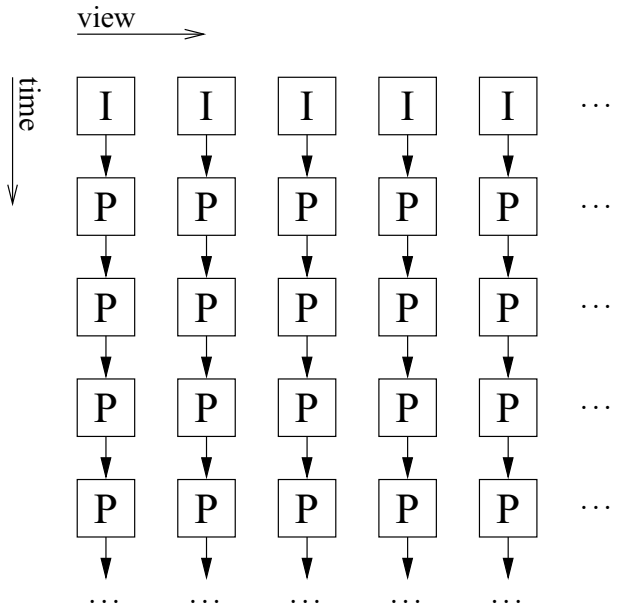

Figure 2: Simulcast Coding

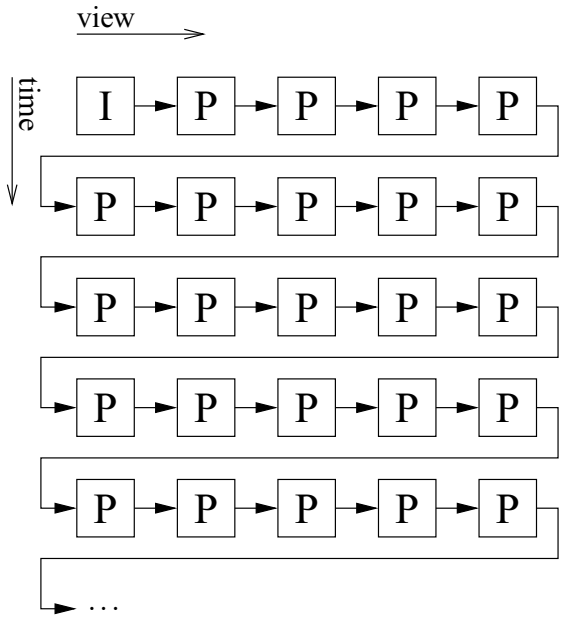

Figure 3: Transposed Picture Ordering

3DAV group within MPEG. One was generated by KDDI Corporation [2]. It consists of several test sequences, each of them captured using a setup of five or eight cameras. The distance between neighbouring camera positions is 20 cm. The image resolution is 320×240 pixels, the frame rate is 30 frames per second. The results described later in section 5 are exemplarily shown for one of the sequences with eight views, "Flamenco1".

Another test dataset is called "Xmas" and was generated by Tanimoto Laboratory, Nagoya University [5]. The dataset is very dense, as the distance between neighbouring capturing positions is only 3 mm, while the distance between the camera and the scene is 300 mm. It contains 101 views of 101 frames each, with a resolution of 640×480 pixels. In fact, the sequence was captured using a single camera only, which leads to good calibration properties, but also to unnatural motion. For the coding experiments described here, only a part of the views was used to reduce the memory requirements of the encoder and decoder as well as the encoding time.

3. SIMULCAST USING H.264/AVC

As a reference, a simulcast scheme was analysed applying the H.264/AVC video coding standard. For that, each camera view of the sequence is coded separately, just like a normal video stream. The first frame of each view is coded as an I-frame, the remaining frames are predictively coded as P-frames (see Figure 2).

As illustrated in Figure 2, a multi-view sequence can be thought of as a matrix in which each element is a picture. The horizontal direction is assigned to the different views, the vertical direction to the time axis. Simulcast coding can then be achieved by independently coding the matrix column by column.

4. TRANSPOSED PICTURE ORDERING FOR IMPROVED PREDICTIVE CODING

In the following, a simple scheme is described which is able to exploit the spatial as well as the temporal redundancy contained in light fields or multi-view sequences. The frames of the sequence are resorted in a way shown in Figure 3. All frames belonging to the first time step are transmitted first. After that, all frames from the second time step are transmitted, and so on.

This can also be regarded as transposing the matrix from Figure 2 and applying simulcast coding again. However, all columns of the transposed matrix are concatenated, and only the very first frame of the sequence is coded as an I-frame. In a practical application, an I-frame would need to be inserted again after a certain time interval. The necessary number of I-frames is however reduced in this scheme compared to simulcast coding.

By using an appropriate number of reference frames in the encoder, not only the spatial correlation can be exploited, but also the temporal correlation. As shown in Figure 4, $N+1$ reference frames are used, where N is the number of views contained in the light field. Therefore, the preceding frame in temporal order as well the preceding frame in spatial order are among the frames which can be used for the prediction.

In this scheme, the memory requirements of the encoder and decoder increase with the number of views. That is why it only works for light fields with a limited number of cameras, as the number of reference frames in the H.264/AVC standard is limited. However, the scheme can be implemented very easily, and besides the process of resorting the pictures, it is possible to use off-the-shelf coding software or hardware.

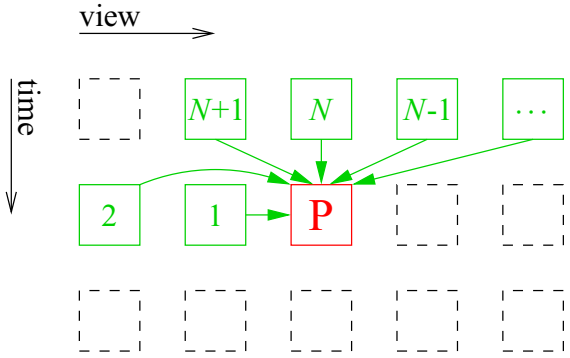

Figure 4: For N views, $N+1$ reference frames are used

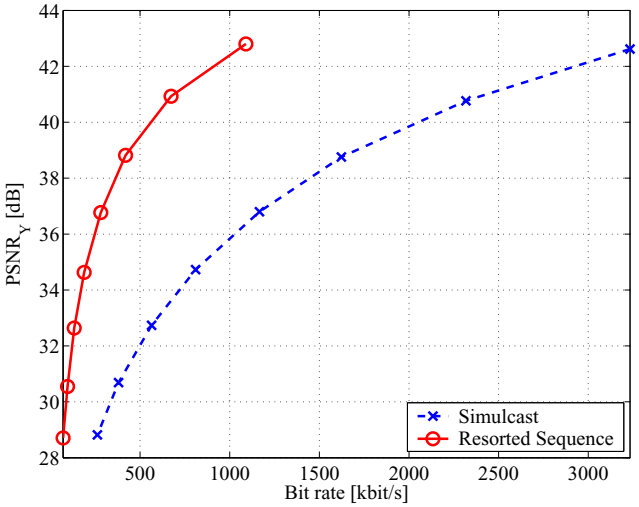

Figure 5: Coding results for the Xmas sequence (8 neighbouring cameras, 101 frames)

5. CODING RESULTS

5.1 "Xmas" Sequence

Figure 5 shows the coding results for the Xmas sequence in the simulcast case and in the case of the resorted sequence. To make the encoding complexity and time comparable, the same number of reference frames has been used for the simulcast case as well as for the resorted sequence. In both cases, the PSNR values plotted in the curves are an average over all frames in all camera views of the decoded dataset.

For this — rather dense — dataset, transposed picture ordering performs clearly better than simulcast coding. The compression factor is about three times higher for the same PSNR value. For a fixed bit rate, a PSNR gain of about 6 dB can be achieved.

In Figure 6, the dependency of the coding performance on the camera distance is shown. For this simulation, the sequence was subsampled in the spatial direction by omitting views. As one would expect, the spatial correlation is lower for higher camera distances, and therefore the coding efficiency deteriorates. As a reference, the performance of the simulcast scheme is also shown, which does not depend on the spatial camera distance. In all cases, the resorted Xmas

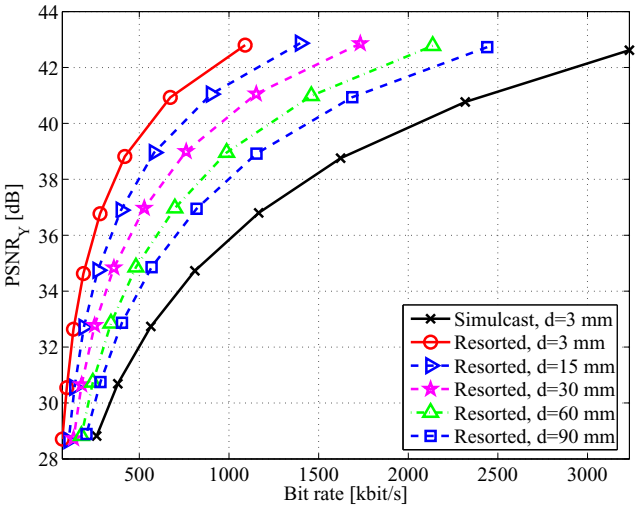

Figure 6: Coding performance for the Xmas sequence with different camera distances d

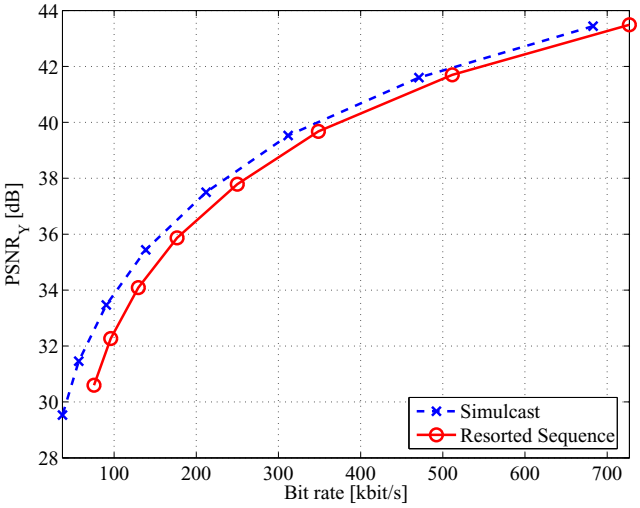

Figure 7: Coding results for the Flamenco1 sequence (8 cameras, 312 frames)

sequence still performs better than simulcast, even for high subsampling factors.

5.2 "Flamenco1" Sequence

Figure 7 shows the coding results for the Flamenco1 sequence. The frame rate of this sequence is 30 frames per second. One can see that transposed picture ordering performs slightly worse than simulcast for this sequence. A possible explanation is that the coder can only use one preceding frame in the temporal direction, while in the simulcast case, it can use up to nine temporal reference frames.

To analyse how the frame rate affects the performance of transposed picture ordering, the sequence was temporally downsampled by omitting frames. The results are shown in Figure 8. If the sequence is downsampled, the original bit rate gets smaller, and therefore the coded bit rate for different frame rates is not a useful indicator for comparing the coding efficiencies. That is why in Figure 8, the bit rate is adjusted as

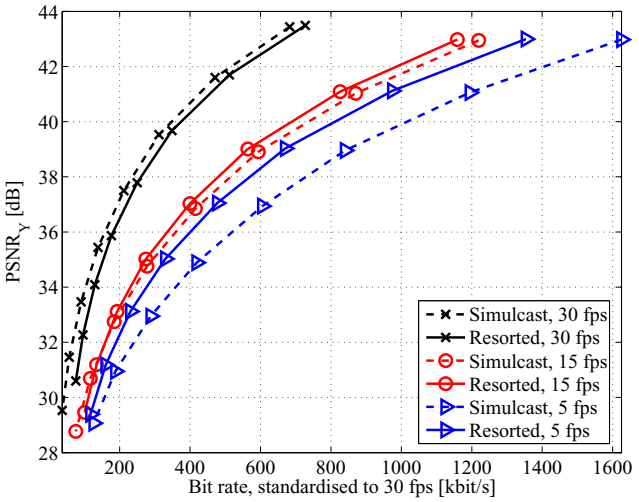

Figure 8: Coding performance for the Flamenco1 sequence with various frame rates (bit rate standardised to 30 fps)

if all the sequences had a frame rate of 30 frames per second.

From this plot, it becomes obvious that the compression factor of both schemes becomes worse for low bit rates because the temporal correlation deteriorates. However, for lower frame rates, transposed picture ordering performs better than simulcast. When the frame rate of the sequence decreases, the gain of the scheme increases.

6. CONSIDERATIONS ON THE POSITIONS OF FRAMES IN THE BUFFER

When more than one reference frame is used in a normal video coder, as it is done for the simulcast coding scheme in Figure 2, the coder searches all these reference frames while motion compensation is performed. It is more likely that the best reference is found in the last frame in the buffer than in frames which are multiple time steps behind the current picture.

That is why in the entropy coding step, shorter codewords are assigned to references with shorter temporal distances to the current block. If rate-distortion optimisation is done, this leads to an even higher probability that the last frame in the buffer is the chosen reference.

For transposed picture ordering, this is however not the optimal solution, because in the resorted sequence, the temporally preceding frame is N frames away from the current frame. This picture is however still very likely to serve as the best reference in the motion compensation step. Therefore, the entropy coding step and rate-distortion optimisation in a classical video coder might not be optimal for the presented scheme, and adaptations in these steps could further improve the performance of the scheme.

7. SUMMARY

Coding results for different classes of multi-view video sequences were presented. As a reference, the performance of the simulcast case using H.264/AVC was investigated, where the video stream from each camera is coded separately. Transposed picture ordering was introduced, where the frames from all cameras are interleaved into a new video stream and the resulting sequence is coded with an H.264/AVC coder. This scheme has the advantage of easy implementation and the possibility to use off-the-shelf coding software or hardware. However, without modifications it can only be used for a rather small number of cameras.

It was shown that for dense datasets like the Xmas sequence, the scheme is able to achieve a significant gain compared to simulcast coding. No gain could however be achieved for sparser datasets like the Flamenco1 sequence. The presented coding results indicate that the gain (or loss) of the scheme strongly depends on the sequence itself, the camera distance and the frame rate. The achieved gain increases when the camera distance becomes smaller or when the frame rate decreases.

Further investigations are necessary on the order of the frames in the buffer, considering that entropy coding and rate-distortion optimisation might not yet be optimal when a standard video coder is applied on the resorted sequences.

REFERENCES

[1] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings SIGGRAPH 96,* pp. 43–54, New Orleans, Louisiana, USA, Aug. 4–9, 1996.

[2] R. Kawada, "KDDI multiview video sequences for MPEG 3DAV use," MPEG document M10533, Munich, Germany, Mar. 2004.

[3] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings SIGGRAPH 96,* pp. 31–42, New Orleans, Louisiana, USA, Aug. 4–9, 1996.

[4] M. Magnor, "Geometry-adaptive multi-view coding techniques for image-based rendering," dissertation, University of Erlangen-Nuremberg, 2000.

[5] M. Tanimoto and T. Fujii, "Test sequence for ray-space coding experiments," MPEG document M10408, Hawaii, USA, Dec. 2003.

[6] M. Tanimoto, *Free Viewpoint Television — FTV,* in *Picture Coding Symposium 2004,* San Francisco, California, USA, Dec. 15–17, 2004.

[7] A. Vetro, W. Matusik, H. Pfister, and J. Xin, "Coding approaches for end-to-end 3D TV systems," in *Picture Coding Symposium 2004,* San Francisco, California, USA, Dec. 15–17, 2004.

[8] C. Zhang and T. Chen, "A survey on image-based rendering — representation, sampling and compression," *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, Jan. 2004.