INCORPORATING DEPTH-IMAGE BASED VIEW-PREDICTION INTO H.264 FOR MULTIVIEW-IMAGE CODING

Yannick Morvan[1], *Dirk Farin*[1] *and Peter H. N. de With*[1,2]

[1] Eindhoven University of Technology, P.O. Box 513
5600 MB Eindhoven, Netherlands
{y.morvan,d.s.farin}@tue.nl

[2] LogicaCMG, RTSE, P.O. Box 7089
5605 JB Eindhoven, Netherlands
P.H.N.de.With@tue.nl

ABSTRACT

We investigate the coding of multiview images obtained from a set of multiple cameras. To exploit the inter-view correlation, two view-prediction tools have been implemented and used in parallel: a block-based motion compensation scheme and a Depth Image Based Rendering technique (DIBR). Whereas DIBR relies on an accurate depth image, the block-based motion-compensation scheme can be performed without any geometry information. Our encoder adaptively selects the most appropriate prediction scheme using a rate-distortion criterion for an optimal prediction-mode selection. The attractiveness of the algorithm is that the compression algorithm is robust against inaccurately estimated depth images and requires only one single reference camera for fast random-access to different views. We present experimental results for several multiview sequences, that result in a quality improvement of up to $1.4\ dB$ as compared to H.264 compression.

Index Terms— Multiview video coding, depth image based rendering, predictive coding, prediction methods, video coding.

1. INTRODUCTION

A 3D video is typically obtained from a set of synchronized cameras, which are capturing the same scene from different view points (multi-view video). This technique enables applications such as free-viewpoint video or 3D-TV. Free-viewpoint video applications provide the feature to interactively select a viewpoint of the scene. With 3D-TV, the depth of the scene can be perceived using a multi-view display that shows simultaneously several views of the same scene. Considering the free-viewpoint video application, random access to neighboring views after coding is necessary so that an appropriate coding structure should be adopted. To exploit both spatial (i.e. inter-view) and temporal redundancy, it has been proposed [1] to use predefined views as a spatial reference from which neighboring views are predicted. Observing the coding structure of Figure 1, it can be seen that temporal correlation is exploited only with respect to the *central* reference view. Similarly, only *non-central* views exploit the spatial inter-view redundancy. For this reason, by exploiting an appropriate mixture of temporal and spatial prediction, views along the chain of cameras can be randomly accessed. Therefore, as opposed to an alternative approach [2], we have adopted the coding structure from Fig. 1 to perform multi-view coding. By doing so, we follow recent suggestions [3] in the 3DAV group within MPEG which indicate alternative prediction structures should be investigated.

A major problem when dealing with multi-view video is the large amount of data to be encoded, decoded and rendered. For example, an independent transmission of eight views of the "Breakdancers" sequence at a PSNR of $40\ dB$ requires about $10\ Mbit/s$. In a typical multi-view acquisition system, the acquired views are highly correlated. As a result, a coding gain can be obtained by exploiting the inter-view dependency between neighboring cameras. To this end, two different approaches for predictive coding of views have been investigated.

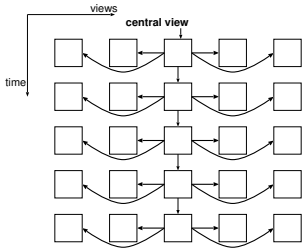

Fig. 1. Coding structure where only the central view employs temporal prediction. This central view is then used as a reference for inter-view prediction.

A first inter-view prediction technique [4] uses a block-based motion-prediction scheme. Besides compatibility with H.264 coding, a major advantage of this approach is that motion compensation does not rely on the geometry of multiple views, so that camera calibration parameters are not required. However, in the case the baseline distance between cameras is high, it has been reported [4] that a block-based motion-compensation scheme yields a limited coding gain over independent coding of the views. One reason is that the translational motion model employed by the block-based motion-compensation scheme is not sufficiently accurate to predict the motion of objects with different depth.

A second, alternative view-prediction scheme [5, 1] is based on an Depth Image Based Rendering algorithm (DIBR). The synthesis algorithm employs a reference texture and depth image as input data. The advantage of the DIBR prediction is that the views can be better predicted even when the baseline distance between the reference and predicted cameras is large, thus yielding a high compression ratio. However, as opposed to the previous approach, the multi-camera acquisition system needs to be fully calibrated prior to the capture session. Additionally, a depth image should be estimated for the central reference view. Because depth estimation is a complicated task, the central depth image may be inaccurately estimated, which thereby reduces the view-prediction quality.

The important requirements of the view prediction algorithm are that (a) it should be robust against inaccurately estimated depth images and (b) an efficient compression should be obtained for various baseline distances between cameras. As discussed above, both pre-

sented view-prediction algorithms have their limitations and cannot be used under variable capturing conditions. Therefore, our novel strategy is to use both algorithms selectively on an image-block basis, depending on their coding performance.

In this paper, we propose a H.264 based multiview encoder that employs the above-described prediction structures using a central reference picture. The view-prediction is performed using both prediction techniques, such that the most appropriate prediction method is selected for each image-block using a rate-distortion criterion. The first view-prediction algorithm is based on block-based motion-prediction. The second view-prediction technique works by warping the reference picture using the corresponding reference depth-image and a DIBR technique, i.e. the *Relief Texture* mapping [6]. We express the relief texture mapping with an alternative formulation that fits better the camera calibration framework. To provide random access to different views, we employ a single picture as a reference from which neighboring views are predicted. While only one reference camera is used for predictive coding, we show that the view-prediction is sufficiently accurate to obtain an efficient compression. To evaluate the efficiency of the prediction across the views, we have integrated the relief-texture-based prediction algorithm into an H.264 encoder. Experimental results show that the proposed prediction algorithm yields up to $1.4\ dB$ improvement when compared to block-based motion prediction using H.264 coding.

The remainder of this paper is organized as follows. Section 2 provides details about the relief-texture view-prediction algorithm while Section 3 shows how the two prediction algorithms can be integrated into an H.264 encoder. Experimental results are provided in Section 4 and the paper concludes with Section 5.

2. PREDICTIVE-CODING OF MULTI-VIEW IMAGES

With block-based motion-prediction [4], the views are multiplexed and the standard H.264 motion-compensation technique is employed.

2.1. Prediction using relief texture mapping

A single texture image and a corresponding depth image are sufficient to synthesize novel views from arbitrary positions. Let us consider a 3D world point $\boldsymbol{P_w} = (X_w, Y_w, Z_w)^T$ captured by two cameras and projected onto the reference as well as the predicted image at *homogeneous* pixel positions $\boldsymbol{p_1} = (x_1, y_1, 1)^T$ and $\boldsymbol{p_2} = (x_2, y_2, 1)^T$, respectively. We assume that the first reference camera is located at the coordinate-system origin and looks along the Z-direction. The location and orientation of the predicted camera are described by its camera center $\boldsymbol{C_2}$ and the rotation matrix $\boldsymbol{R_2}$. This allows us to define the pixel positions $\boldsymbol{p_1}$ and $\boldsymbol{p_2}$ in both image planes by

$$\lambda_1 \boldsymbol{p_1} = \boldsymbol{K_1}(X_w, Y_w, Z_w)^T , \quad (1)$$

$$\lambda_2 \boldsymbol{p_2} = \boldsymbol{K_2 R_2}(X_w, Y_w, Z_w)^T - \boldsymbol{K_2 R_2 C_2} , \quad (2)$$

where $\boldsymbol{K_1}$, $\boldsymbol{K_2}$ represent the 3×3 intrinsic parameter matrix of the corresponding cameras and λ_1, λ_2 some positive scaling factors [7]. Because the matrix $\boldsymbol{K_1}$ is upper-triangular and $\boldsymbol{K_1}(3,3) = 1$, the scaling factor λ_1 can be specified in this particular case by $\lambda_1 = Z_w$. From Equation (1), the 3D position of the original point $\boldsymbol{P_w}$ in Euclidean coordinates can be written as

$$(X_w, Y_w, Z_w)^T = \boldsymbol{K_1}^{-1} \lambda_1 \boldsymbol{p_1} = \boldsymbol{K_1}^{-1} Z_w \boldsymbol{p_1}. \quad (3)$$

Finally, we obtain the predicted pixel position $\boldsymbol{p_2}$ by substituting Equation (3) into Equation (2), so that

$$\lambda_2 \boldsymbol{p_2} = \boldsymbol{K_2 R_2 K_1}^{-1} Z_w \boldsymbol{p_1} - \boldsymbol{K_2 R_2 C_2}. \quad (4)$$

Equation (4) constitutes the image-warping [8] equation that enables the synthesis of the predicted view from the original reference view and its corresponding depth image.

One issue of the previously described method is that input pixels $\boldsymbol{p_1}$ of the reference view may not always be mapped to a pixel $\boldsymbol{p_2}$ at an integer pixel position. A second difficulty is that multiple original pixels can be projected onto the same pixel position in the predicted view. For example, a foreground pixel can occlude a background pixel in the interpolated view, which is resulting in overlapping pixels. Additionally, some regions in the interpolated view are not visible from the original viewpoint, which results in holes in the predicted image. To address the aforementioned issues, we describe a variant of the *relief texture* mapping technique that we have adapted to the geometry of multiple views.

The guiding principle of the relief texture algorithm is to factorize the 3D image-warping equation into a combination of 2D texture mapping operations. One well-known 2D texture mapping operation corresponds to a perspective projection of planar texture onto a plane defined in a 3D world. Mathematically, this projection can be defined using homogeneous coordinates by a 3×3 matrix multiplication, and corresponds to an homography transform between two images. The advantage of such a transformation is that a hardware implementation of this function is available in most of the Graphic Processor Units (GPU). Processing time is therefore dramatically reduced. Let us now factorize the warping function so as to obtain a homography transform in the factorization. From Equation (4), it follows that

$$\frac{\lambda_2}{Z_w} \boldsymbol{p_2} = \boldsymbol{K_2 R_2 K_1}^{-1} \cdot (\boldsymbol{p_1} - \frac{\boldsymbol{K_1 C_2}}{Z_w}). \quad (5)$$

Analyzing this equation, it can be seen that the first factor $\boldsymbol{K_2 R_2 K_1}^{-1}$ is equivalent to a 3×3 matrix and represents the desired homography transform.

Let us now analyze the second factor of the factorized equation, i.e. $(\boldsymbol{p_1} - \boldsymbol{K_1 C_2}/Z_w)$. This second factor projects the input pixel $\boldsymbol{p_1}$ onto an intermediate point $\boldsymbol{p_i} = (x_i, y_i, 1)^T$ that is defined by

$$\lambda_i \boldsymbol{p_i} = \boldsymbol{p_1} - \frac{\boldsymbol{K_1 C_2}}{Z_w}, \quad (6)$$

where λ_i defines a homogeneous scaling factor. It can be seen that this last operation performs the translation of the reference pixel $\boldsymbol{p_1}$ to the intermediate pixel $\boldsymbol{p_i}$. The translation vector can be expressed in homogeneous coordinates by

$$\lambda_i \begin{pmatrix} x_i \\ y_i \\ 1 \end{pmatrix} = \begin{pmatrix} x_1 - t_1 \\ y_1 - t_2 \\ 1 - t_3 \end{pmatrix} \text{ with } (t_1, t_2, t_3)^T = \frac{\boldsymbol{K_1 C_2}}{Z_w}. \quad (7)$$

Written in Euclidean coordinates, the intermediate pixel position is defined by

$$x_i = \frac{x_1 - t_1}{1 - t_3} \qquad y_i = \frac{y_1 - t_2}{1 - t_3}. \quad (8)$$

It can be seen that this result basically involves a 2D texture mapping operation, which can be further decomposed into a sequence of two 1D-transformations. In practice, these two 1D-transformations are performed first along rows, and then along columns. This class of warping methods is known as scanline algorithms [9]. An advantage

of this additional decomposition is that a simpler 1D texture mapping algorithm can be employed (as opposed to 2D texture mapping algorithms).
For the padding of occluded pixels, we have employed simple heuristic techniques where occluded pixels are padded by adjacent background pixels [10].

As an algorithmic summary, the synthesis of the view using relief texture mapping is performed as follows.

- Step 1: Perform warping of reference texture along horizontal scanlines.
- Step 2: Perform warping of the (already horizontally-warped) texture along vertical scanlines.
- Step 3: Compute the planar texture projection of the intermediate image using the homography transform defined by $\boldsymbol{K_2 R_2 K_1^{-1}}$ (exploiting the GPU for fast computing).

3. INCORPORATING RELIEF TEXTURE INTO H.264

In this section, we describe our novel H.264 architecture dedicated to multi-view coding that employs a block-based motion-prediction scheme and the relief-texture-mapping image-warping technique.

One approach to integrate both to integrate both prediction techniques, warping-based prediction and block-based motion prediction, would be to select the better prediction for each block. However, the prediction can be improved by performing a warping-based prediction followed by a motion-prediction on the same block (see Figure 2). The system concept becomes as follows. First, we provide an approximation of the predicted view using relief texture mapping and, second, we refine the warping-based prediction using block-based motion prediction. In the refinement stage, the search for matching blocks is performed in a region of limited size, e.g. 32×32 pixels. In contrast to this, the disparity between two views in the "Ballet" sequence can be as high as 50 pixels. Figure 2 shows an overview of the described coding architecture.

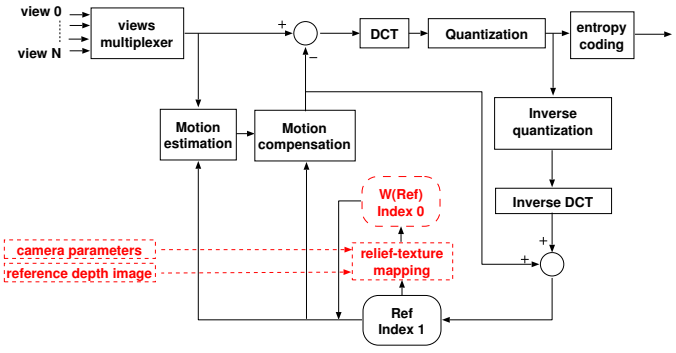

Fig. 2. Architecture of an H.264 encoder that adaptively employs a block-based motion prediction or relief-texture image-warping prediction followed by a prediction-refinement. The central reference frame and the corresponding warped reference frame are denoted Ref and $W(Ref)$, respectively.

The advantages of using an H.264 encoder are many-fold. First, re-using a standardized encoder provides forms of backward compatibility with the H.264 compression functions (CABAC, etc.). Second, because the H.264 standard enables that each macroblock can be encoded using different coding modes, occluded regions in the predicted view can be efficiently compressed. More specifically, occluded pixels cannot always be predicted with sufficient accuracy. In this case, the algorithm encodes an occluded macroblock in *intra-mode*. Alternatively, when the prediction accuracy of occluded pixels is sufficient, the macroblock is encoded in *inter-mode*. Third, in the case that the depth image is not estimated accurately, the image-warping prediction is simply not selected. Finally, the prediction mode is specified for each image-block to the decoder by the reference frame index. Thus, the H.264 standard offers sufficient flexibility in coding modes to match them with the various prediction accuracies of our algorithm.

4. EXPERIMENTAL RESULTS

For evaluating the performance of the coding algorithm, experiments were carried out using the "Ballet" and "Breakdancers" sequences. The presented experiments investigate the impact of the prediction accuracy on the rate-distortion performances, using the prediction structure depicted by Figure 1. For each presented rate-distortion curve (Figure 3), we perform the compression of multi-views under two different conditions.

1. The prediction of views is carried out using only the H.264 block-based motion-prediction.
2. The prediction of views is carried out adaptively, enabling also the warping-based prediction described in this paper.

To ensure that in our evaluation the motion-prediction over time does not interfere with the evaluation of the inter-view prediction algorithm, each reference view is encoded as an intra-frame.

For coding experiments, we employed the open-source H.264 encoder x264 [11]. The arithmetic coding algorithm CABAC was enabled for all experiments and the motion search was 32×32 pixels. All predicted frames are encoded as P-frames while reference texture and depth frames are encoded as I-frames. We set the number of reference frames to 2: one reference for the block-based motion-prediction and a second for the warping-based prediction. Prior to warping, the reference depth is encoded with quantizer setting $QP = 29$. It should be noted that depth images should be encoded at a relatively high quality to avoid ringing-artifacts along object borders in the depth map. This prevents that rendering artifacts occur in the warping-based predicted view. This remark is similar to the conclusions related to recent depth compression results [12]. A dedicated depth image coder has been previously developed by the author to prevent these specific rendering artifacts [13].

Because depth data is necessary for 3D rendering in any case, it can be assumed that depth images are transmitted even in the case no warping-based prediction is employed. Hence, employing the warping-based prediction does not involve any bit-rate overhead. It should therefore be noted that the presented rate-distortion curves in Figure 3 do not include the bit-rate of depth images.

Let us now discuss the obtained rate-distortion curves of Figure 3(a) and Figure 3(b). First, it can be observed that the proposed warping-based prediction algorithm consistently outperforms the block-based motion-prediction scheme. For example, considering Figure 3(a), the warping-based prediction algorithm yields a quality improvement of up to $1.4\ dB$ at $1.8\ Mbit/s$ over the block-based motion-prediction algorithm. Considering Figure 3(b), despite predicted views show large regions of occluded pixels, up to $1\ dB$ quality improvement was obtained at a bit-rate of $2\ Mbit/s$.

In Fig. 4, it can be seen that occluded image-blocks at the right side of the dancer are intra-coded. Moreover, we observe that the relief-texture-based prediction performs efficient prediction within

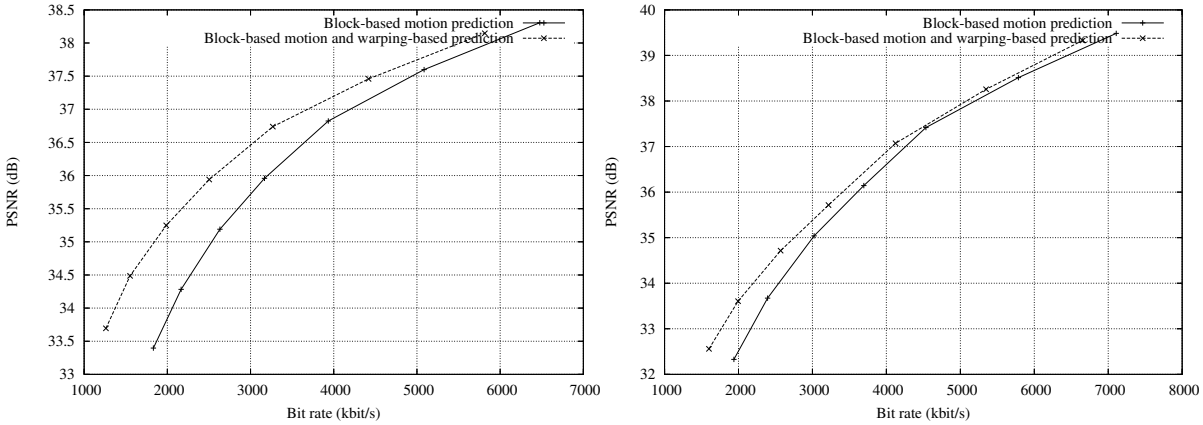

Fig. 3. Rate-distortion curves for encoding 8 views of (a) the "Breakdancers" and (b) the "Ballet" sequences.

untextured areas. In this case, the image-block is mostly coded as a Skip mode. However, we observed that the warping-based prediction cannot always perform accurate prediction in textured blocks. In this case, the block-based motion prediction is selected.

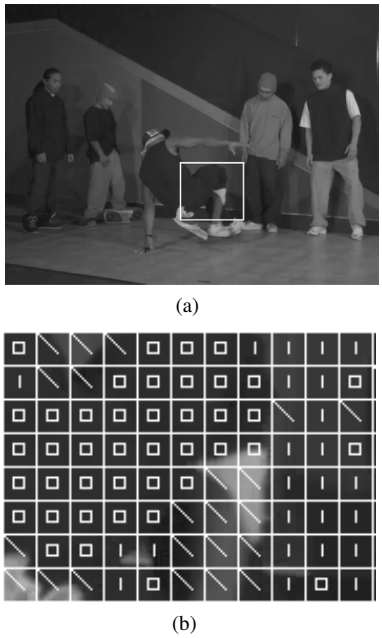

Fig. 4. (a) A coded view 2 of the sequence "Breakdancers" (b) Magnified view of the marked area, coding modes "intra", "block-based motion prediction" and "warping-based prediction" are indicated by a vertical line, a diagonal line and a square respectively.

5. CONCLUSIONS

We have presented an algorithm for the predictive coding of multiple camera views that employs two different view-prediction algorithms: a block-based motion prediction and a warping-based prediction. The advantages of the algorithm are that the compression is robust against inaccurately estimated depth images and that the chosen prediction structure allows random access to different views. For each image-block, the selection between the two prediction algorithms is carried out using a rate-distortion criterion. The warping-based prediction employs the relief texture mapping algorithm which can be efficiently executed on a GPU. We express the relief texture mapping with an alternative formulation that fits better the camera calibration framework. Furthermore, we have integrated the prediction scheme into an H.264 encoder, such that motion-compensation prediction is combined with the warping-based prediction. Experimental results have shown that the warping-based predictive-coding algorithm can improve the resulting image quality by up to $1.4\ dB$ when compared to solely performing H.264 block-based motion prediction.

6. REFERENCES

[1] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 600–608, 2004.

[2] E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of h.264/avc for multiview video compression," in *IEEE Int. Conf. on Image Proc.*, Atlanta, USA, October 2006.

[3] A. Vetro, "CE10: View synthesis prediction for MVC," ISO/IEC JTC1/SC29/WG11 and ITU SG16 Q.6, 2006.

[4] P. Merkle, K. Mueller, A. Smolic, and T. Wiegand, "Efficient compression of multi-view video exploiting inter-view dependencies based on H.264/MPEG4-AVC," in *Int. Conf. on Mult. and Expo, ICME 2006*, Toronto, Canada, 2006, vol. 1, pp. 1717–1720.

[5] M. Magnor, P. Ramanathan, and B. Girod, "Multi-view coding for image based rendering using 3-D scene geometry," *IEEE Trans. on CSVT*, pp. 1092–1106, November 2003.

[6] M. M. Oliveira, *Relief Texture Mapping*, Ph.D. Dissertation. UNC Computer Science, March 2000.

[7] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2004.

[8] L. McMillan, *An Image-Based Approach to Three-Dimensional Computer Graphics*, University of North Carolina, April 1997.

[9] G. Wolberg, *Digital Image Warping*, IEEE Computer Society Press, July 1990.

[10] Y. Morvan, D. Farin, and P. H. N. de With, "Design considerations for view interpolation in a 3D video coding framework," in *27th Symp. on Inf. Theory in the Benelux*, 2006.

[11] "Webpage title: x264 a free H264/AVC encoder," http://developers.videolan.org/x264.html, last visited: December 2006.

[12] C. Fehn, N. Atzpadin, M. Mller, O. Schreer, A. Smolic, R. Tanger, and P. Kauff, "An advanced 3DTV concept providing interoperabilty and scalabilty for a wide range of multi-baseline geometries," in *IEEE Int. Conf. on Image Proc.*, Atlanta, USA, October 2006.

[13] Y. Morvan, P. H. N. de With, and D. Farin, "Platelet-based coding of depth maps for the transmission of multiview images," in *Stereoscopic Displays and Applications XVII, Proceedings of the SPIE*, 2006.