Multi-view Video Coding using View Interpolation and Reference Picture Selection

Masaki Kitahara[a], Hideaki Kimata[b], Shinya Shimizu[a], Kazuto Kamikura[a], Yoshiyuki Yashima[a], Kenji Yamamoto[c], Tomohiro Yendo[c], Toshiaki Fujii[c], Masayuki Tanimoto[c]

[a]Cyber Space Laboratories, NTT Corporation
[b]NTT Advanced Technology Corporation
[c]Department of Electrical Engineering and Computer Science,
Graduate school of Engineering, Nagoya University

ABSTRACT

We propose a new multi-view video coding method using adaptive selection of motion/disparity compensation based on H.264/AVC. One of the key points of the proposed method is the use of view interpolation as a tool for disparity compensation by assigning reference picture indices to interpolated images. Experimental results show that significant gains can be obtained compared to the conventional approach that was often used.

1. INTRODUCTION

A multi-view video is a group of videos captured by multiple cameras of the same scene. Free Viewpoint Television (FTV) and 3DTV (display technologies that enable depth perception for the viewer) are known to be the main applications for multi-view video, which have gained a large amount of interest in the industry as new types of visual entertainment applications. For this reason, capturing, processing, and coding multi-view video has become a highly active reaserch topic. Furthermore, after the establishment of an adhoc group called "3DAV" in MPEG [1], MPEG has recently determined to make a standard on multi-view video coding.

When coding Multi-view video, the key technical point is how to jointly utilize temporal, spacial, and inter-view (between cameras) correlation. A well known approach is to apply block-based adaptive selection of intra coding, motion/disparity compensation-based coding [2], [3], [4]. Most of these methods apply block-based motion compensation-like disparity compensation (ie. encode block-based 2D disparity vectors and prediction residuals), with block partitioning such as ones in H.264/AVC. Although these approaches have shown to be effective, we point out that by exploiting properties specific to multi-view video, we could expect better inter-view prediction, and as a result, gain higher coding efficiency.

Based on this standpoint, we consider exploitation of view interpolation as a prediction tool for multi-view video coding. View interpolation is a term for techniques that generate images of novel views from images captured from multiple cameras (which is also termed "image based rendering" in some literatures [5]). To be specific, we consider the use of ray space method [6], [7], [8], which the quality of the interpolated image has been well discussed.

In section 2, we review the method that the proposed approach is based on.In section 3, we explain how view interpolation is incorporated into the coding framework explained in section 2. Experimental results are shown in section 4, and conclusions are given in section 5.

2. DISPARITY COMPENSATION BY REFERENCE PICTURE SELECTION AND OUR CURRENT IMPLEMENTATION

The proposed coding scheme is based on our previously proposed approach [2] which enables high coding efficiency by allowing block-based adaptive selection of intra coding, motion/disparity compensation-based coding. In this paper, disparity compensation does not only mean prediction by reference to a decoded image of a different camera corresponding to the same time instant, but also prediction by reference to decoded images of a different camera of a different time instant. Basic components of I, P, and B frame coding of H.264/AVC are used. For I frames, H.264/AVC without any modification is used. For P and B frames, block-based intra-frame coding and inter-frame predictive coding (we refer to "inter-frame predictive coding" as coding by motion/disparity compensation) can be selectively applied for each block of each frame by reference picture selection described below. Most of the syntax elements including prediction residuals are coded as in H.264/AVC (see [2] for details).

Consider camera indices $c = 0, 1, .., C$ which are indices assigned to each camera, and a set of camera indices $\boldsymbol{C}_c \subset \{0, 1, .., C\}$ which are indices to cameras that camera c references to for disparity compensation . Upper part of figure 1 illustrates one example of a prediction structure and reference camera indices. In the following, we will refer to cameras which do not reference other cameras (coded by H.264/AVC without modification) as base cameras, and cameras which reference other camera as inter cameras. Reference camera indices $\boldsymbol{C}_c$ must be encoded/sent to the decoder, so that reference picture index assignment explained below can be done.

In our current implementation, coding order of frames depends on prediction structure among cameras and the GOP structure used for the base camera. Figure 1 shows an example, where squares indicates frames and number on the right is the coding order. camera $c = 0$ has IBBP GOP structure and is coded in the same way as in H.264/AVC. After encod-

ing IBBP for camera $c = 0$, frames of other cameras of the same time interval are encoded in the same order as camera $c = 0$.

Each camera has decoded picture buffers (DPBs), and these DPBs operate in the same way as in H.264/AVC. In other words, most currently decoded images are stored for each camera. However, when encoding a frame for an inter camera c, DPBs that correspond to reference camera indices $\boldsymbol{C}_c$ and DPB of camera c are shared. And before coding a frame, list 0 and list 1 reference picture indices are assigned to all the decoded images in the shared DPBs at that time instant. In our current implementation, a default assignment rule is defined which does not need to be signaled to the decoder. Optionally, different assignments can be used by signaling the assignment information to the decoder. This can be done for each slice and is coded by universal variable length coding (UVLC) and is embedded in the slice header. As in H.264/AVC, reference picture selection can be done for 16x16, 16x8, 8x16, 8x8 macroblock partitioning, and motion compensation technique in H.264/AVC is applied for the selected reference picture (which means that blocks of size 4x4 at the smallest can have motion/disparity vectors). Since these reference indices include indices to decoded frames for other cameras, disparity compensation is adaptively applied.

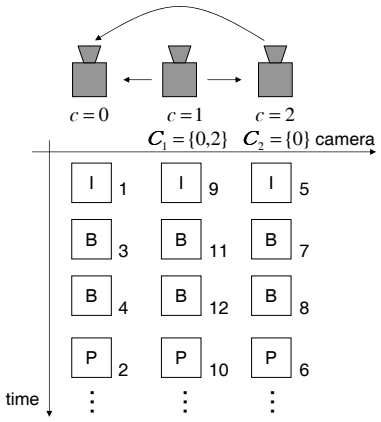

Fig. 1. Example of the coding order in our current implementation

3. VIEW INTERPOLATION-BASED DISPARITY COMPENSATION

3.1. View interpolation-based disparity compensation by reference picture selection procedure

For inter cameras which reference two or more cameras, we propose a method to incorporate view interpolation as a prediction method for disparity compensation. View interpolation is done using decoded images of two cameras corresponding to the same picture order count. In order to do that, we also assign reference picture indices to images generated by view interpolation along with assigment to decoded images in the DPBs. When a reference picture index assigned to interpolated image is selected of a block, the encoder/decoder generates an interpolated block and uses it for diparity compensation. The residual with respect to the original image is encoded. This residual is coded by the same method as motion compensation residual coding in H.264/AVC. Processing of disparities will be dicussed in the next section.

The advantages of assigning reference picture indices to interpolated images are the following. Using conventional rate-distortion optimization methods [9], adaptive selection of appropriate prediction methods (motion compensation, disparity compensation by using decoded images directly, and disparity compensation by view interpolation) can be done. Since neither of the prediction method is optimal for every part of the image, robustness can be gained by this flexiblity that can be optimized at the encoder. More importantly, since we use entropy coding method in H.264/AVC to encode reference picture indices, reference picture index values close to zero are represented with smaller bits. Thus, with the mechanism explained in section 2 to re-assign reference picture indices, adaptive assignment can be done depending on the bit rate, properties of the muti-view video etc.

In our current implementation, for generating view interpolation for an inter camera c, the pair of cameras that can be used is selected from cameras corresponding to reference camera indices $\boldsymbol{C}_c$. If $\boldsymbol{C}_c$ includes more than 2 cameras, we can consider more two or more pairs. Thus, if maximum pairs of cameras is given by $maxNumPairs$, reference picture indices can be assigned to $maxNumPairs$ interpolated images at maximum.

3.2. Interpolation by disparity estimation and block-based correction

In order to perform view interpolation, camera parameters for all rellavent cameras are needed which are assumed to be available at the encoder/decoder. Furthermore, our current implementation assumes that all the cameras are arranged in a single line with equal distance between cameras, and the optical (z) axes, the x axes, and the and y axes are all parallel as illustrated in figure 2. Since in practice, such exact alignment of camera is not possible, reference pictures are corrected so that the optical axis is parallel, using camera parameters prior to view interpolation. In the following, we denote reference pictures as corrected pictures. In order to generate a view interpolated image of a block subject to encoding, we need per-pixel diparities to the two reference pictures.

In the follwing, we denote (m, n) as pixel coordinates (horziontal and vertical respectively), the two cameras that are used for reference as c_0 and c_1, and the horizontal (x axes) displacements of the two cameras from the camera subject to encoding as M_0 and M_1.Firstly, disparities can be represented solely as horizontal pixel differences, due to epipolar con-

straint. Secondly, for a pixel m, n in the frame to be encoded, if the disparity to camera c_0 is given as $d(m, n)$ in pixels, disparity to camera c_1 is given as $(d(m, n)M_1)/M_0$ pixels. This also means that with the knowledge of the camera parameters, disparity to both camera can be derived if disparity to one of the camera is given.

The encoder of our proposed scheme, first estimates the disparities corresponding to all pixels (m, n) of the block subject to encoding, without using the block to be encoded (ie. using only the reference pictures of cameras c_0 and c_1). This can be accomplished by any of the method described in [6], [7], [8]. According to these methods, this is done by minimizing an error energy function with respect to $d(m, n)$, which is a function of the error between the two blocks in reference pictures of cameras c_0 and c_1 with the central pixels as $(m + d(m, n), n))$ and $(m + (d(m, n)M_1)/M_0, n))$. Note that $d(m, n)$ can be computed at the decoder also without sending any overhead from the encoder because $d(m, n)$ are computed solely from reference pictures.

In many cases, by using the estimated disparities $d(m, n)$, an interpolated image of certain degree of quality can be genarated. However, since estimated disparities $d(m, n)$ are not optimized with respect to the block subject to encoding, we observed that there is significant amount of prediction error energy in some cases. Furthermore, it is not straight forward to apply rate-distortion optimization, where for high bit rates it is important to reduce prediction error energy (rather than reducing overhead bits for sending disparity to the decoder), and where for low bit rates it is the opposite. A straight forward solution to this problem is to send correction values with respect to the estimated disparities $d(m, n)$. However, sending correction values per-pixel will spend significant amount of bits. Instead, in our proposed scheme, we take the middle-ground, where only one scalar correction value is sent per block (subject to encoding). As with motion vectors in H.264/AVC, a correction value can be applied for 4x4 block at the smallest. Thus the amount of correction values that are encoded can be adaptively controlled by rate-distortion optimization (ie. macroblock mode selection). Denoting e as the correction value, the actual disparities used for interpolation for a given block is given as follows.

$$\acute{d}(m, n) = d(m, n) + e \quad (1)$$

As noted before, with $\acute{d}(m, n)$, disparities to camera c_1 can be derived.

In our current implementation, correction value e is determined by minimizing an error function with respect to e, which is a function of the actual prediction error (differences of interpolated image and the block subject to encoding). Denoting the reference picture of camera c_0 and c_1 as $I_0(m, n)$ and $I_1(m, n)$ respectively, interpolated image $P(m, n)$ is simply computed by average of corresponding pixels, as given below.

$$P(m, n) = I_0(m + \acute{d}(m, n), n)/2 + I_1(m + (\acute{d}(m, n)M_1)/M_0, n)/2 \quad (2)$$

The correction value e is encoded losslessly by entropy coding method of H.264/AVC (CABAC) for motion vectors with minor modification to the context modeling method in our current implementation.

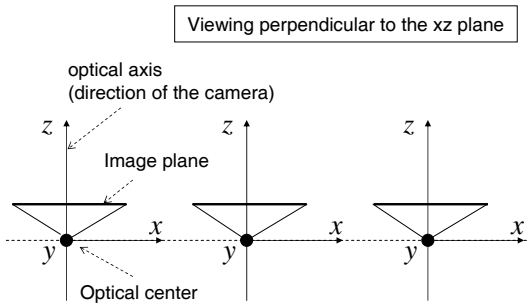

Fig. 2. Illustration of the assumed camera arrangement

4. EXPERIMENTS

4.1. Encoding strategies/conditions

In order to assess the effect of using view interpolation, we compared our coding results by the method overviewed in section 2 and our proposed scheme that combines view interpolation to the method in 2.

The data used was "Rena" sequence which was provided by the co-authors of Nagoya University (Tanimoto laboratory) to MPEG as test data for evaluation in the standardization precess [10]. This data was originally captured by 100 cameras. Each video is corrected so that we can consider these video as captured by camera arrangement in figure 2, and are VGA (640x480) each consisting of 300 frames.

Camera $c = 43$ to $c = 45$ was encoded to show example results by prediction structure given in figure 3. Video of camera $c = 42$ and $c = 46$ was encoded by method overviewed in section 2 prior to this experiment, and the decoded images were used as reference pictures.

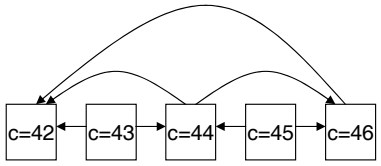

Fig. 3. Prediction structure for "Rena"

Since our implementation is based on the H.264/AVC reference software JM9.5, rate-distoriton optimization scheme

used in this software was used with minor modifications. Rate control is not used, where coding by fixed quantization parameter was applied.

In order to assign appropriate reference picture indices to interpolated images, an adhoc reference picture assignment was done by the following preliminary experiment. We first encoded these videos using the default reference picture index assignment. Frequency of usage for each reference picture index value with respect to every frame was aquired from this encoding. We re-assigned the reference picture index corresponding to interpolated image by this frequency data for each frame, by using the per-slice re-assignment method described in section 2. In these preliminary experiments, we observed that at lower bitrates, smaller reference picture index value was re-assigned to interpolated images.

4.2. Results

Figure 4 illustrate example rate-distortion characteristics comparison. Although for camera $c = 43$ a degradation was observed, for other two cameras, it can be seen that significant coding gains were obtained. Furthermore, there is a tendency to obtain gains at lower bitrates where PSNR values are around 37 to 40 dB, where quality differences are generally noticeable to the human eye. The degradation for camera $c = 43$ can be of many reasons including high inaccuracy of camera parameters.

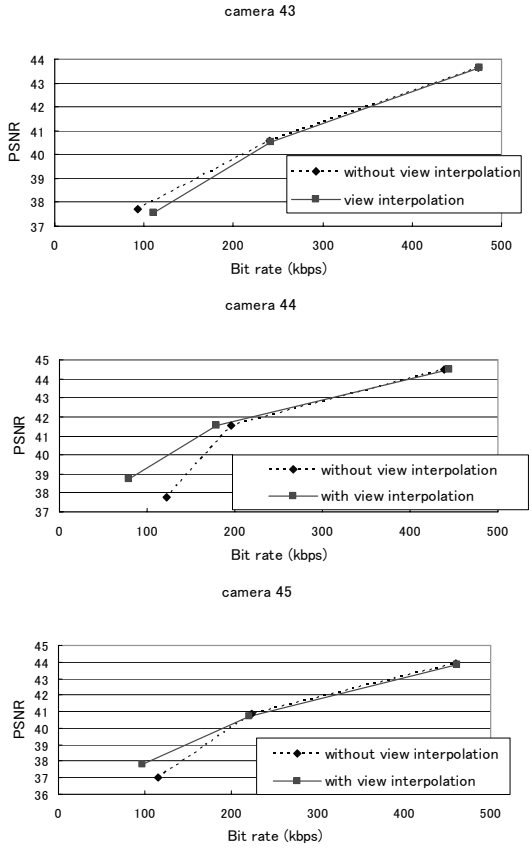

Fig. 4. Experimental results

5. CONCLUSION

We have proposed a multi-view video coding scheme based on the H.264/AVC framework which utilize both temporal/inter-view correlation by adaptive use of motion/disparity compensation. The key point of our approach was the application of view interpolation for disparity compensation by assigning reference picture indices to interpolated images. From experimental results, it was shown that significant additional coding gains can be obtained.

6. REFERENCES

[1] "Report on 3dav exploration," document N5878 MPEG Trondheim Meeting, 2003.

[2] H. Kimata, M. Kitahara, K. Kamikura, and Y. Yashima, "Multi-view video coding using reference picture selection for freeviewpoint video communication," *PCS2004*, 2004.

[3] S. C. Chan, K. T. Ng, Z. F. Gan, K. L. Chan, and H. Y. Shum, "The plenoptic video," *IEEE Trans., Circuits Syst., Video Technol.*, vol. 15, no. 12, pp. 1650– 1659, 2005.

[4] "Survey of algorithms used for multi-view video coding (mvc)," document N6909 MPEG Hong Kong Meeting, 2005.

[5] H. Y. Shun, S. B. Kang, and S. C. Chan, "Survey of image-based representations and compression techniques," *IEEE Trans., Circuits Syst., Video Technol.*, vol. 13, no. 11, pp. 1020–1037, 2003.

[6] T. Kobayashi, T. Fujii, T. Kimoto, and M. Tanimoto, "Interpolation of ray-space data by adaptive. filtering," *IS&T/SPIE Electronic Imaging*, pp. 252–259, 2000.

[7] M. Droese, T. Fujii, and M. Tanimoto, "Ray-space interpolation based on filtering in disparity domain," *3D Image Conference 2004*, pp. 213–216, 2004.

[8] M. Droese, T. Fujii, and M. Tanimoto, "Ray-space interpolation constraining smooth disparities based on loopy belief propagation," *IWSSIP2004*, pp. 247–250, 2004.

[9] G. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, pp. 74–90, Nov. 1998.

[10] "Call for proposals on multi-view video coding," document N7327 MPEG Poznan Meeting, 2005.