RENDERING-ORIENTED DECODING FOR DISTRIBUTED MULTI-VIEW CODING SYSTEM

Yuichi Taguchi and Takeshi Naemura

Graduate School of Information Science and Technology, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan
{yuichi, naemura}@hc.ic.i.u-tokyo.ac.jp

ABSTRACT

This paper discusses a system in which multi-view images are captured and encoded in a distributed fashion and a viewer synthesizes a novel view from this data. We developed an efficient method for such system that combines decoding and rendering process to directly synthesize the novel image without reconstructing all the input images. Our method jointly performs disparity compensation in decoding process and geometry estimation in rendering process, because they are essentially equivalent if the camera parameters for the input images are known. It achieves low-complexity for both encoder and decoder in distributed multi-view coding system. Experimental results show superior coding performance of our method compared to a conventional intra-coding method especially at low bit rate.

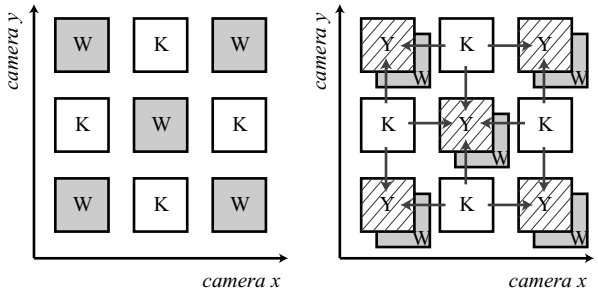
***Index Terms*—** Data compression, Image coding, Rendering (computer graphics), Stereo vision

1. INTRODUCTION

Camera array systems can capture multi-view images of a 3D scene, which allow a viewer to observe the scene from arbitrary viewpoints with image-based rendering techniques [1]. Such systems require efficient coding schemes owing to the large amount of data, typically consisting of hundreds of views. Since they capture an identical scene from slightly different viewpoints, significant correlations exist among the multi-view images. Most of conventional methods exploit the correlations at the encoder using the concept of disparity compensation. However, this requires high encoding complexity and the communication between cameras with large data volume.

Distributed multi-view coding schemes provide a solution for such problems [2–5]. In these methods, each image is encoded independently, but decoded jointly at a central decoder. Since the inter-camera communication is avoided, low-complexity encoding and simple system configuration can be achieved. The inter-image correlation is exploited at the decoder. Therefore, the compression efficiency is still higher than conventional intra-coding method. In previous works, however, the decoder seems to pay unnecessary computational cost when the viewer only observes a novel image; that is, it first reconstructs input camera images and then synthesizes the novel image with a general renderer targeting the decoded images. To our knowledge, there is no approach so far that synthesizes a novel image directly from the encoded data.

We consider a system in which multi-view images are captured and encoded in a distributed fashion, and a remote viewer synthesizes a novel image at a desired viewpoint using this data. We propose an efficient method that combines decoding and rendering process so that the novel image can be directly synthesized without reconstructing all the input images. This rendering-oriented decoding method jointly performs two key techniques: disparity compensation in decoding process and geometry estimation in rendering process, since they are essentially equivalent if the camera parameters for the multi-view images are known. When the viewer only synthesizes a novel image, our method requires low computational cost compared to the typical method that performs above two processes separately. Our method achieves low-complexity for both encoder and decoder as a conventional intra-coding method, while shows better coding performance due to the inter-image decoding.



(a) Encoder (b) Decoder

Fig. 1. A typical structure of distributed multi-view coding system.

2. BACKGROUND

2.1. Distributed Multi-View Coding

Figure 1 shows a typical structure of distributed multi-view coding. The images are classified into two categories: key images (K) and Wyner-Ziv images (W). The key images are encoded and decoded independently with a conventional intra-image coder. The Wyner-Ziv images are encoded independently with a channel coder, and their parity bits are transmitted to the decoder. To decode the Wyner-Ziv image, its estimate called side information (Y) is generated through disparity compensated prediction using the previously decoded key images, and the prediction error is corrected using the parity bits of the image.

The coding efficiency of the above method greatly depends on the accuracy of the side information, because only a few parity bits are needed to correct small prediction error. If the scene geometry is available, accurate side information can be generated by warping the neighboring views [3]. For multi-view video sequences, motion compensated prediction can be combined with disparity compensated one to improve the quality of side information [4, 5].

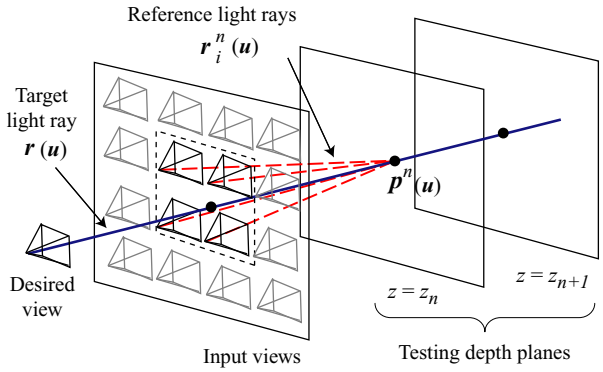


Fig. 2. Depth estimation method for synthesizing a desired view.

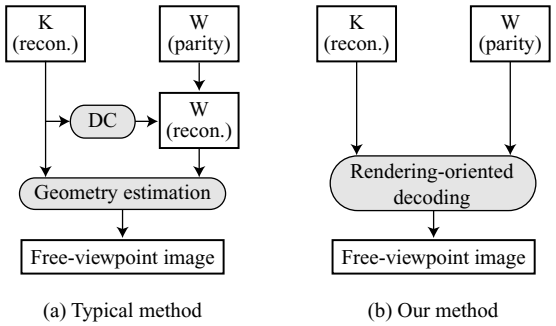


Fig. 3. Process flow for synthesizing a free-viewpoint image. (DC: Disparity compensation)

2.2. Rendering with Multi-View Images

Suppose that multi-view images are captured with many cameras that roughly lie on a plane and are arranged in a 2D grid, and that there is no prior knowledge of the scene geometry. To synthesize a novel image from this data, geometry estimation is widely adopted to compensate for the lack of cameras and interpolate the data appropriately [6, 7]. Here we consider an on-the-fly estimation method of the depth map depending on the desired viewpoint.

As shown in Fig. 2, a layered depth model, $z_n (n = 1, 2, .., N)$, is assumed in the object space. We estimate the depth for each target light ray, $\boldsymbol{r}(\boldsymbol{u})$, where $\boldsymbol{u}$ represents the position of the light ray in the desired view. At the intersection of the target light ray with each of depth layers ($\boldsymbol{p}^n(\boldsymbol{u})$), we evaluate the similarity (color consistency [6] or focus measure [7]) of the reference light rays, which correspond to the back-projections of the intersection point to the input cameras and are denoted by $\boldsymbol{r}_i^n(\boldsymbol{u})$, where i is a camera index. To prevent the occlusion effect and keep computational cost low, this similarity evaluation is often performed using only the k-nearest cameras [6–8]; therefore its cost function is given by

$$J(\boldsymbol{p}^n(\boldsymbol{u})) = similarity\,(I(\boldsymbol{r}_i^n(\boldsymbol{u})|_{i\in V})), \tag{1}$$

where V is the set of camera indices near the target light ray, and $I(\cdot)$ denotes the color of the light ray. In our implementation, $|V| = k = 4$ as shown in Fig. 2. For reducing the noise effect, this cost function is smoothed in each depth layer. Finally, the depth of each target light ray is selected by

$$n(\boldsymbol{u}) = \arg\min_n J(\boldsymbol{p}^n(\boldsymbol{u})), \tag{2}$$

and its color is given by the average color of the reference light rays

$$I\,(\boldsymbol{r}(\boldsymbol{u})) = average\,(I(\boldsymbol{r}_i^{n(\boldsymbol{u})}(\boldsymbol{u})|_{i\in V})). \tag{3}$$

3. RENDERING-ORIENTED DECODING

The rendering method described above can be used if all images (to be more accurate, image segments [9]) needed to synthesize the desired view are reconstructed and available; therefore, as shown in Fig. 3(a), typical methods first reconstruct the multi-view images with the decoding method described in Section 2.1. However, they seem to pay unnecessary computational cost, since disparity compensation in decoding process and geometry estimation in rendering process are essentially equivalent if the camera parameters for the multi-view images are known.

To synthesize a desired view directly, we propose rendering-oriented decoding method, in which the decoding of the Wyner-Ziv images is incorporated into the rendering process, as shown in Fig. 3(b). The Wyner-Ziv images are therefore not reconstructed explicitly. Our method uses a simple coset code for the Wyner-Ziv images. It achieves low-complexity for both encoder and decoder as a conventional intra-coding method.

3.1. Rendering Algorithm with Coset Codes

The input multi-view images are divided into key images and Wyner-Ziv images. At the encoder, the key images are encoded using a conventional intra-image coder. For the Wyner-Ziv images, the pixel value is represented by M cosets, $C_m (m = 1, 2, .., M)$, in a memoryless fashion [10].

At the decoder, we first reconstruct the key images and coset indices for the Wyner-Ziv images. The side information for each target light ray and each depth layer, $Y^n(\boldsymbol{u})$, is then calculated by averaging the color of the reference light rays in key images

$$Y^n(\boldsymbol{u}) = average\,(I(\boldsymbol{r}_i^n(\boldsymbol{u})|_{i\in V_K})), \tag{4}$$

where V_K is the set of camera indices for the key images in V. Using this side information, we reconstruct the reference light rays of near Wyner-Ziv images in a maximum likelihood sense by

$$\hat{I}(\boldsymbol{r}_i^n(\boldsymbol{u})|_{i\in V_W}) = \arg\min_{c_j\in C_m} (c_j - Y^n(\boldsymbol{u}))^2, \tag{5}$$

where V_W is the set of camera indices for the Wyner-Ziv images in V, and c_j is a codeword in the coset C_m for the light ray $\boldsymbol{r}_i^n(\boldsymbol{u})|_{i\in V_W}$. We then evaluate the similarity of the reference light rays by Eq. (1) and estimate the depth and color for each target light ray by Eqs. (2) and (3). Since the extra computational cost for Eqs. (4) and (5) is not too large, we can keep the complexity of this rendering method as low as that of the original one described in Section 2.2.

3.2. Implementation

Figure 4 shows the implementation diagram of our proposed method. The key images are encoded with discrete wavelet transform (DWT) and SPIHT [11], implemented in QccPack [12], for each RGB component. For the Wyner-Ziv images, we first map each RGB value of the pixel to a coset by the function shown in Fig. 5 [13]. The coset indices are then encoded with the DWT and SPIHT as well as the key images. Since we use the lossy coder for the coset indices, we choose the mapping function shown in Fig. 5, instead of the regular modulo M function, to prevent drastic changes in codewords with a

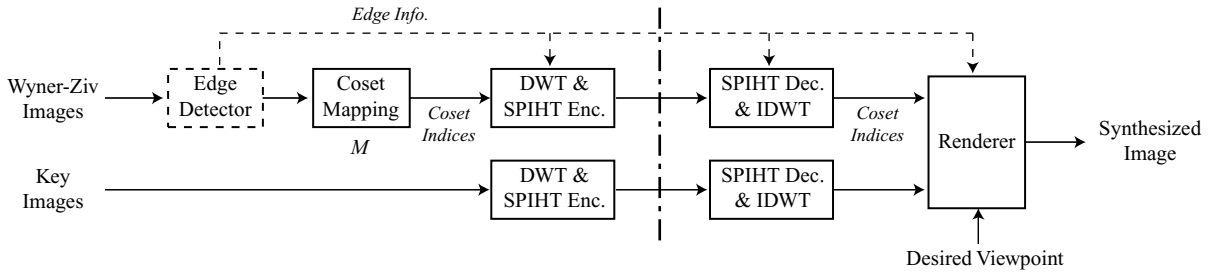


Fig. 4. Implementation diagram.

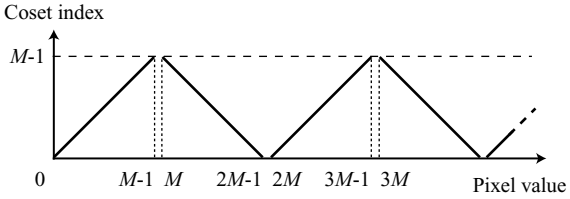


Fig. 5. Coset mapping function.

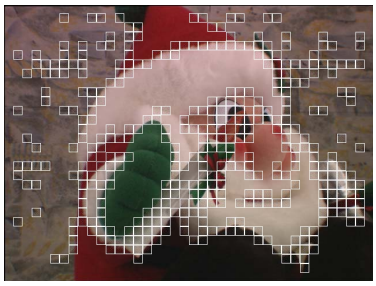


Fig. 6. Extracted edge regions in an input image of the *Doll* data set.

small error of the coset index. At the decoder, we decode the SPIHT and perform the rendering-oriented decoding with the decoded key images and coset indices of the Wyner-Ziv images.

In generation of the side information for the Wyner-Ziv images, smooth regions can be easily predicted, while edge regions are difficult due to the occlusion. In other words, the predicted color by Eq. (4) is enough accurate in the smooth regions, while it includes large error in the edge regions [5]. Therefore, we also implemented a simple edge detector for the Wyner-Ziv images to improve the compression efficiency. The variance of image segment of 16×16 pixels is evaluated to extract the edge regions. Figure 6 shows the extracted edge regions in our experiments. To encode only the edge regions, we use shape adaptive SPIHT [12] with the mask image for the edge regions. The smooth regions have no information in this implementation. Hence, the correction procedure by Eqs. (4) and (5) is performed only for the edge regions.

4. EXPERIMENTS

The complexity of our method is almost as low as that of the method encoding all images as the key images and synthesizing a novel image with the normal renderer described in Section 2.2, which is referred as all key method. We therefore compared the coding performance of them in this experiment, to show the advantage of our method.

We used the *Doll* image set provided by courtesy of University of Tsukuba, Japan, as shown in Fig. 6. This data set consists of 81 (9×9) still images of 640×480 pixels, which are captured with cameras arranged in a regular 2D grid on a plane. We divided these images into 41 key images, which are referred as base-K images hereafter, and 40 Wyner-Ziv images as shown in Fig. 1; therefore $|V_K| = |V_W| = 2$ for all target light rays. The color variance of the reference light rays was used for the similarity evaluation in Eq. (1).

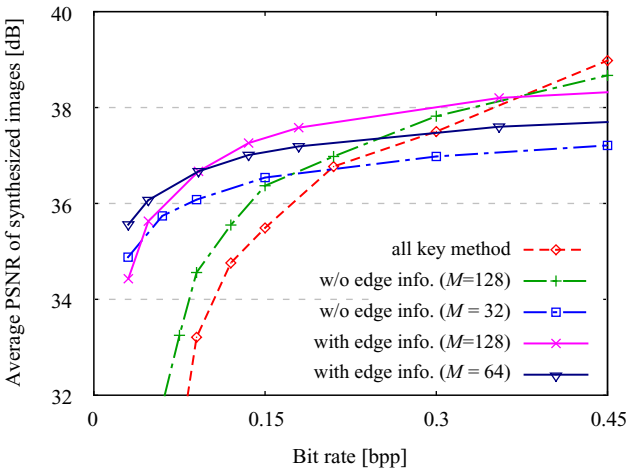
Figure 7 shows the rate-distortion performance of our method either with or without the edge detector, compared to that of the all key method. We fixed the bit rate of the base-K images to 0.3 bpp and 0.9 bpp, where the average quality of them was 34.04 dB and 40.98 dB as peak signal-to-noise ratio (PSNR), for Figs. 7(a) and (b), respectively. The bit rate of the other images (Wyner-Ziv images for our method and key images for the all key method) was controlled by truncating the SPIHT bitstream and expressed on the horizontal axis. The plots show the reconstruction quality of synthesized images averaged for random 10 viewpoints, where the quality is calculated with respect to the image synthesized from the uncompressed data and expressed as PSNR. The bit rate of edge information is included in the plot of our method using it.

As it can be seen from the plots, our proposed method shows superior performance especially at low bit rate. Smaller M yields better performance at low bit rate, because small error in the smooth regions can be corrected by the coset code with small M, but it restricts the maximum quality which is important at high bit rate. Since we do not use feedback channel to control the rate for the Wyner-Ziv images [3, 4], it is still a difficult problem to decide proper M at the encoder for efficient rate control. The edge information provides additional gain for our method at low bit rate, since the edge regions include larger error than the smooth regions.

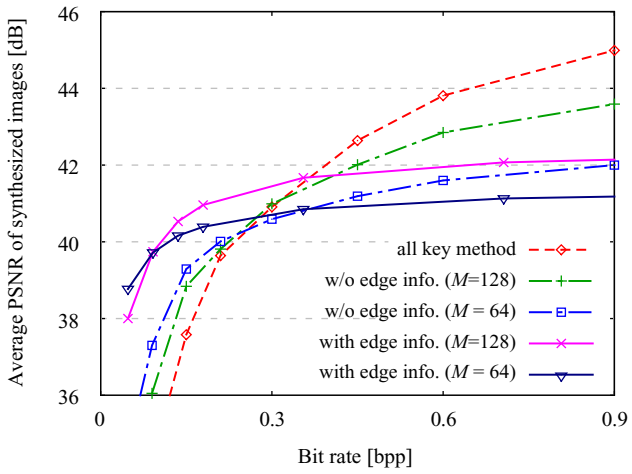
Figure 8 shows the reconstructed synthesized images using the all key method and our method with edge information. The bit rate of base-K images is 0.3 bpp and that of the other images is 0.15 bpp. For the all key method, it can be observed that the edge regions have large error as shown in Fig. 8(a). Our method with edge information decreases the error in the edge regions as shown in Fig. 8(b).

5. CONCLUSIONS

In this paper, we proposed the rendering-oriented decoding method for distributed multi-view coding system. Our method directly synthesizes a novel image without reconstructing the Wyner-Ziv images explicitly, by incorporating the reconstruction of light rays in the Wyner-Ziv images into the rendering process. It achieves both low-complexity encoder and decoder as a conventional intra-coding method, while shows better coding performance especially at low bit rate. Future work will be focused on investigating an estimation method for determining appropriate number of cosets at the encoder, and extending this method to multi-view video sequences.

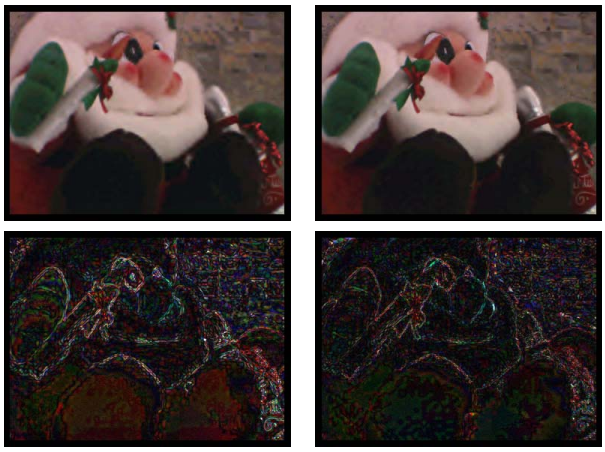


(a) With low-quality base-K images (0.3 bpp, 34.04 dB)



(b) With high-quality base-K images (0.9 bpp, 40.98 dB)

Fig. 7. Rate-distortion curves.



(a) All key method
35.45 dB

(b) Our method with edge info.
37.55 dB (*M*=128)

Fig. 8. Synthesized images and their difference from that using uncompressed data (multiplied by 8).

Acknowledgment: We wish to acknowledge valuable discussions with Prof. H. Harashima and Dr. K. Takahashi at the University of Tokyo, Japan.

6. REFERENCES

[1] H.-Y. Shum, 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, Nov. 2003.

[2] A. Jagmohan, A. Sehgal, and N. Ahuja, "Compression of light-field rendered images using coset codes," in *Proc. Asilomar Conf. Signals, Systems, and Computers*, Nov. 2003, vol. 1, pp. 830–834.

[3] A. Aaron, P. Ramanathan, and B. Girod, "Wyner-Ziv coding of light fields for random access," in *Proc. IEEE Int. Workshop on Multimedia Signal Processing (MMSP 2004)*, Sept. 2004, pp. 323–326.

[4] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, "Distributed multi-view video coding," in *Proc. SPIE Visual Communications and Image Processing (VCIP 2006)*, Jan. 2006, vol. 6077.

[5] Z. Jin, M. Yu, G. Jiang, X. Zeng, and Y.-D. Kim, "ROI-based Wyner-Ziv coding with low encoding complexity for wireless multiview video sensor array," in *Proc. Picture Coding Symposium (PCS 2006)*, Apr. 2006.

[6] C. Zhang and T. Chen, "A self-reconfigurable camera array," in *Proc. 15th Eurographics Symposium on Rendering*, June 2004, pp. 243–254.

[7] K. Takahashi and T. Naemura, "Layered light-field rendering with focus measurement," *EURASIP Signal Processing: Image Commun.*, vol. 21, no. 6, pp. 519–530, July 2006.

[8] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen, "Unstructured lumigraph rendering," in *Proc. ACM SIGGRAPH 2001*, Aug. 2001, pp. 425–432.

[9] Y. Taguchi, K. Takahashi, and T. Naemura, "View-dependent scalable coding of light fields using ROI-based techniques," in *Proc. SPIE Three-Dimensional TV, Video, and Display V*, Oct. 2006, vol. 6392.

[10] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 626–643, Mar. 2003.

[11] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, June 1996.

[12] "QccPack — quantization, compression, and coding library," http://qccpack.sourceforge.net/.

[13] R. Bernardini, R. Rinaldo, P. Zontone, D. Alfonso, and A. Vitali, "Wavelet domain distributed coding for video," in *Proc IEEE Int. Conf. Image Processing (ICIP 2006)*, Oct. 2006, pp. 245–248.