Efficient Light Field Image Coding with Depth Estimation and View Synthesis

Takanori Senoh
School of Science and
Technology for Future Life
Tokyo Denki University
Tokyo, Japan
senoh-taka@jcom.home.ne.jp

Kenji Yamamoto
Applied Electromagnetic
Research Institute
National Institute of Information
and Communications Technology
Koganei, Japan
k.yamamoto@nict.go.jp

Nobuji Tetsutani
School of Science and
Technology for Future Life
Tokyo Denki University
Tokyo, Japan
tetsutani@mail.dendai.ac.jp

Hiroshi Yasuda
Tokyo Denki University
Tokyo, Japan
mpegyasuda@mail.dendai.ac.jp

***Abstract*— Efficient light field image coding method is proposed based on conversion to multi-view image, depth estimation, multi-view coding, and view synthesis. Firstly, compatibility of light field image and multi-view image is discussed, and then depth estimation method based on texture-edge-aware horizontal and vertical view matching and depth-smoothing is explained. Secondly, a view-synthesis method from up to four reference views is proposed, which adopts depth-base occlusion hole inpainting. Finally, by combining these methods together with a hierarchical bi-directional inter-view coding of multi-view image and depth maps, coding results are reported.**

Keywords—light field, sub-aperture, multi-view, depth estimation, inter-view prediction, view synthesis, depth-base inpainting.

I. Introduction

Light field images are expected to be a promising immersive media, which provides naked-eye 3D scene or immersive augmented reality. Light field image consists of large number of elemental images, which determines image resolution. Number of pixels in each elemental image determines viewing-zone angle of 3D image. Consequently, it requires huge number of pixels to realize a wide viewing-zone angle and high resolution 3D image. For real application, this huge amount of data is the first priority to be solved. Direct compression of light-field image is not efficient, since elemental image array has large high-frequency component. Multi-view image is another representation of light field image and it is reported to be efficiently compressed[1]. It is also reported that multi-view images can be further compressed by using depth maps[2]. According to these works, this paper proposes an improved method of light field image coding based on lossless conversion of light field image to multi-view image, depth estimation from up to four reference views, inter-view prediction coding, and view synthesis from up to four reference views. Section 2 discusses compatibility between light field image and multi-view image. Section 3 explains high-quality edge-aware horizontal and vertical depth estimation from up to four reference views. Acceleration of depth estimation is also discussed for real-time applications. Section 4 proposes a high-quality view synthesis method using up to four reference views and adopting depth-base occlusion hole inpainting technology. Section 5 reports results of the combination of these tools together with fast and efficient multi-view image and depth separate-coding, which adopts a hierarchical bi-directional inter-view prediction. Section 6 concludes this paper.

II. Compatibility of Light Field and Multi-View

A. Light Fied Image

Concept of light field was firstly discussed by Adelson in 1991 as Plenoptic Function[3] which is described in (1).

$$P = P\,(\theta, \Phi, \lambda, t, x, y, z) \qquad (1)$$

where $(\theta,\ \Phi)$ are horizontal and vertical angle, λ is wavelength (color), t is time, and (x, y, z) are location of light in 3D space. This light field can be captured by light field camera, which has convex lens-let array in front of image sensor as shown in Fig. 1[4]. Here, light is sampled at the center of each lens-let (x, y). Since image sensor is 2D plane, z is common and discarded. Light angles $(\theta,\ \Phi)$ are sampled by pixel (u, v) in elemental image covered by each lens-let (x, y). Now, light field image is expressed in (2).

$$P = P\,(x, y, u, v) \qquad (2)$$

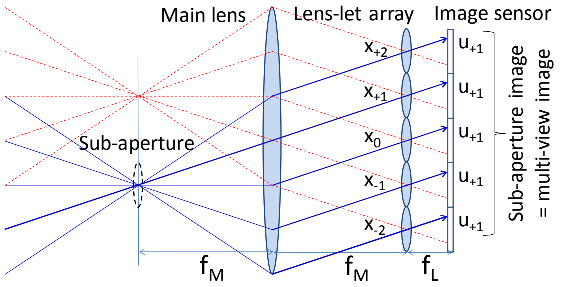


Fig. 1. Ray-trace of light field camera[4]

Color (λ) is sampled by three primary colors (r, g, b) and converted to YUV420. Time (t) is sampled by frame. Since array of elemental images includes large high-frequency component at every edge of elemental images as shown in Fig.2 left, its direct compression is not efficient. Although correlation between elemental images is high, direct multi-view coding of huge number of elemental images will be practically difficult. Currently available light field camera of low resolution will eases this inefficiency for a while.

ISBN 978-90-827970-1-5 © EURASIP 2018

B. Conversion of Light Field Image to Multi-View Image

Main lens is necessary to get enough disparity for far objects and to enable conversion of light filed image to true multi-view image. When main lens exists, since light field image consists of interleaved sub-aperture images which are identical to multi-view image as shown in Fig. 1, light field image can be lossless-converted to multi-view image by taking same-location pixels from all elemental images. Since multi-view images are highly correlated spatially and to each other, efficient compression is expected.

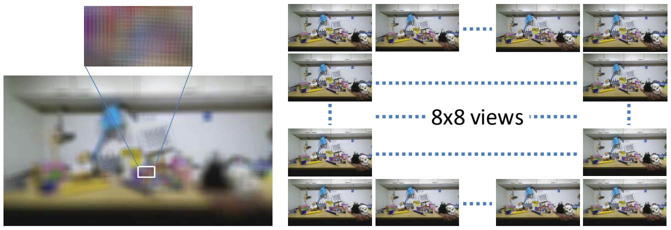


Fig. 2. 1920×1080 pel light field image (left) and its compatible 8×8-view representaion of sub-aperture (mult-view) images (right)[4]

It should be noted that light field images can be captured by 2D multi-view camera array, which provides higher resolution and wider motion parallax than single light field camera. If no main lens exists, converted multi-view like images become parallel projection of 3D scene, which look weird since our eye is perspective projection. Moreover, existing tools for depth estimation or view synthesis do not work appropriately, since they are designed for perspective projection according to human visual system.

III. Depth Estimation

A. Depth Estimation Reference Software

If depth maps (disparity between views) are available for multi-view images, we can reduce the number of views[5]. Discarded views can be synthesized from remaining views with their depth maps. Depth maps are estimated from multi-view image by searching disparity (shift of corresponding pixel location) which minimizes matching error between views. In order to get correct depth map, rectification (coordinate alignment) and color correction of multi-view images are essential. OpenCV[6] provides such function. Furthermore, to avoid occlusion hole problem, where no corresponding pixel exists, three views (left, center, right) are generally used for matching error calculation as illustrated in Fig. 3.

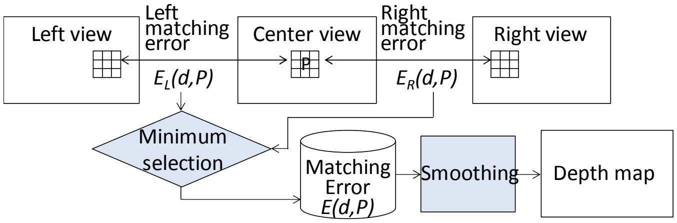


Fig. 3. Depth estimation scheme from three views[5]

Depth maps obtained by just minimizing matching error include many depth errors as shown in Fig. 4 middle left, since matching error is not reliable where image texture is flat. In order to reduce depth errors, Graph-Cuts algorithm is widely used, which minimizes energy of matching error plus neighbor depth difference by testing all depth levels pixel by pixel in the image. MPEG is providing 1D depth estimation reference software (DERS)[7]. Since this algorithm iterates energy minimization until convergence, depth estimation time is long and unpredictable. Fig. 4 also shows 1D multi-view images (top) and estimated depth maps by DERS with Graph-Cuts smoothing (others).

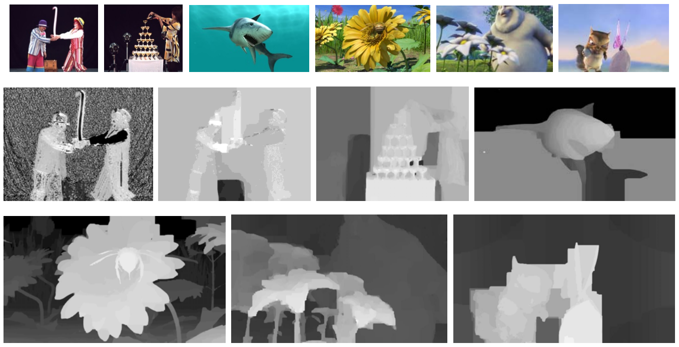


Fig. 4. 1D multi-view images (from top left: Pantomime, Champagne Tower, Shark, Bee, Flowers, Butterfly), depth maps estimated by matching error only (middle left) and with Graph-Cuts smoothing (others)[7][8]

B. Accerelation of Depth Estimation

Acceleration of depth estimation is essential for real-time application. There are some reports which accelerate depth estimation by replacing Graph-Cuts algorithm with non-iterative methods[9][10]. [9] check depth candidate from the lowest level to the highest level pixel by pixel once, and determines whether to keep old level or replace it. It generates texture edge amp with simple high-pass filter (HPF) first. Next, it starts zig-zag scan from middle line, where many reliable matching errors exist at texture edges. Zig-zag scan reduces foreground depth level over-propagation to background area. Where texture edge exists, it reduces depth continuity weight to avoid estimation error. Its depth estimation speed is about 100 times faster than Graph-Cuts smoothing (about 10 sec for 1920×1080 pixels by 3GHz CPU). Estimated depth map quality is comparable to Graph-Cuts smoothing. [10] determines reliable depth levels on texture edge from matching error only. Estimated depth errors are eliminated by 8-directional median filter and dominant filter. Non-edge area is inpainted with lower depth level of left or right texture edge. Its depth estimation time is about 10-200 times faster than Graph-Cuts smoothing with the same depth map quality.

C. 2D Depth Estimation with Reliability and Edge Weighting

Since multi-view images converted from light field image form 2D array as shown in Fig. 2 right, horizontal and vertical view matching will improve depth quality. We enhanced DERS to up to four reference views and added following reliability weight on matching error (*1*) and depth smoothing weight (*2*), as shown in Fig. 5 (eDERS)[11]. Results are in Fig. 5

right. Estimated depth quality was about 2dB higher than 1D depth estimation (DERS).

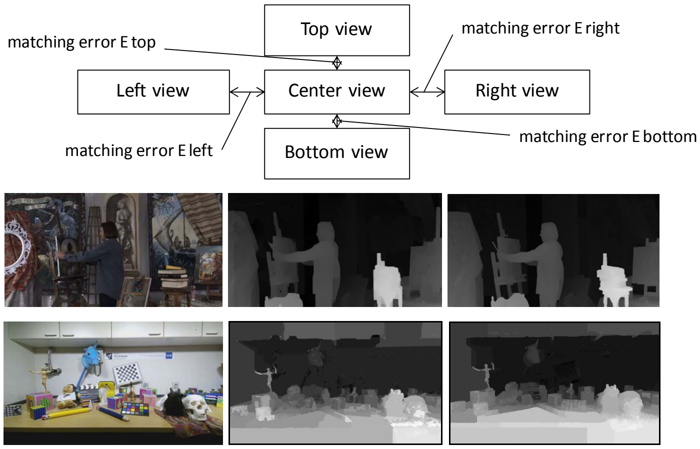


Fig. 5. 2D depth estimation (top), estimated depth by 1D DERS (center), and by 2D eDERS (right) from 2D multi-view image (left: Painter, Unicorn)[11][12]

(1) Reliability weight

For the best error selection from up to four matching errors E (left, right, top, and bottom), reliability weight W is multiplied to raw matching error E, which avoids pseudo-matching. Reliability weight W is determined by measuring horizontal and vertical texture edge slope S as follows.

$$\begin{aligned}
&S = |P\,(p\text{-}1) - P\,(p\text{+}1)|\\
&\text{if } (S < min)\; W = th1\,/\,min\\
&\text{if } (min <= S < th1)\; W = th1\,/\,S \qquad (3)\\
&\text{if } (S >= th1)\;\; W = 1\\
&\text{if } (S = 0\;\&\; E = 0)\; E = th1\,/\,min
\end{aligned}$$

Where, $P(p\text{-}1)$ and $P(p\text{+}1)$ are colors of left and right neighbor pixels, or top and bottom neighbor pixels. $th1$ is given by depth estimation parameter. min = 0.3. S is a weighted average in 3x3-pixel block. Fig.6 shows reliability weight curve (left) and reliability map (right). Dark area is reliable (W = 1, $S >= th1$) and white area is unreliable ($W = th1\,/\,min$, $S < min$).

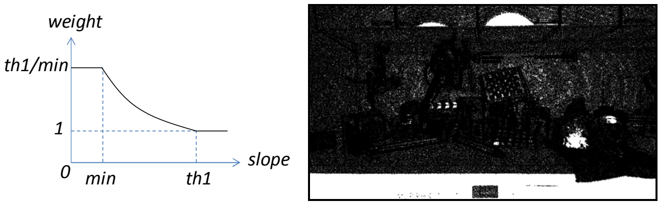


Fig. 6. Reliability weight (left) and reliability map (right) for matching[11]

(2) Depth smoothing weight

In this experiment, depth maps are smoothed by Graph-Cuts algorithm. In order to stop over-propagation of foreground depth level to background area as seen in Fig.4, depth continuity weight is automatically reduced by multiplying texture edge weight ρ. Texture edge is detected by the 2nd order differential of texture color as follows. Where Si = edge slope.

$$\begin{aligned}
&S1 = P\,(p\text{-}1) - P\,(p\text{-}3),\;\; S2 = P\,(p\text{+}1) - P\,(p\text{-}1)\\
&S3 = P\,(p\text{+}3) - P\,(p\text{+}1) \qquad (4)\\
&\text{if}(S2 > th2\;\&\; S2 >= S1, S3\,) \text{ or } (S2 < \text{-}th2\;\&\; S2 <= S1, S3)\\
&\;\; edge = true
\end{aligned}$$

Slope Si are checked in four directions in 7x7-pixel block as shown in Fig. 7 left. ρ and $th2$ are given by depth estimation parameters. Fig. 7 right shows detected edge map.

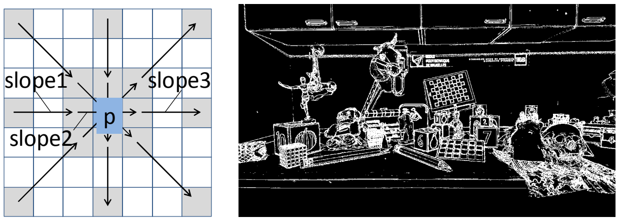


Fig. 7. Edge detector (left) and detected edge map (right) for smoothing[11]

IV. View Synthesis

A. View Synthesis Reference Software

MPEG is also providing a 1D view synthesis reference software (VSRS) [5], which synthesizes in-between views from left and right reference views with their depth maps. Since major synthesis noise comes from occlusion hole where no pixel is available in reference view, two reference views (left, right) are used as shown in Fig. 8. In order to avoid background pixels replacing foreground pixels in virtual view, left and right depth maps are first projected to virtual view (virtual depth L, R), keeping foreground pixels remain. Small depth holes caused by disparity stepping are removed by median filter. Next, virtual depth L and R project left and right views to virtual view (virtual view L, R). Projected virtual view L and R are merged to one virtual view with inter-view distance weight.

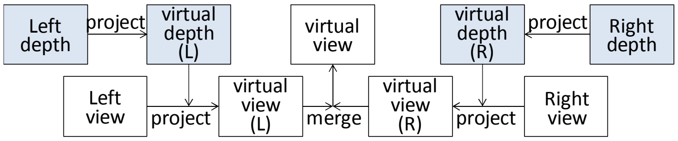


Fig. 8. Views synthesis shema from two reference views [5]

Even though, common holes still remain where no pixel is available from any reference view. Such holes must be inpainted with background pixels since occlusion hole always exists in background area[13]. If no consideration is paid, inpainted holes look like "fat ghost" or "skinny ghost". VSRS4.2 adopted following hole inpainting tool (Ivsrs) proposed in [14]. Since "skinny ghost" is coming from foreground object edges having background depth levels, dilation of occlusion hole solved this problem. Since "fat ghost" is caused by inpainting holes with foreground pixels, it was removed by inpainting holes with background pixels. Besides that, if color difference in reference view exists, edges of hole filled with different reference view become noticeable. These hole edges were smoothed by applying a simple low-pass filter on these hole edges.

B. 2D View Synthesis from up to four Reference Views

We propose following 2D view synthesis method to improve over-all image quality for light-field image coding. In proposed light-field image coding system, some views must be synthesized as shown dark in Fig. 11 top, since they are discarded at encoder side. Instead of using just two reference views, it will be better to use four reference views, top left, top right, bottom left, and bottom right views as shown in Fig. 9 top. We newly enhanced VSRS4.2 to up to four reference views (eVSRS). Fig. 9 bottom shows PSNR of synthesized view by current VSRS4.2 (left) and by new eVSRS (right). eVSRS increased average PSNR of synthesized view by about 0.6dB and reduced PSNR fluctuation over views.

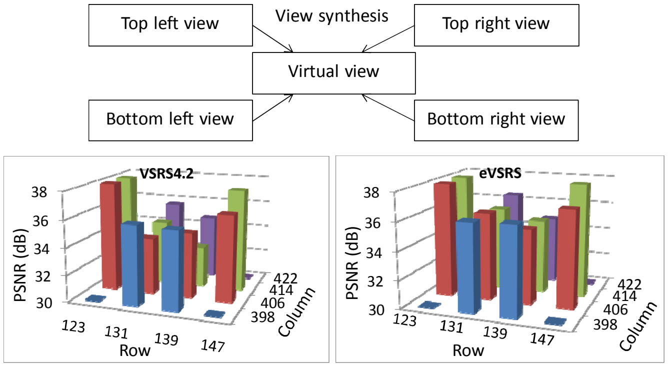


Fig. 9. 2D View synthesis from up to four reference views (top), PSNR of synthesized view by 1D VSRS4.2 (bottom left), by 2D eVSRS (bottom center), and synthesized view by eVSRS (bottom right)

V. Multi-View Coding

A. 1D Multi-View Image Coding with Depth Map

If depth maps are available, some views don't have to be encoded since they can be synthesized after decoding. We proposed such coding schema (fDEVS) [15] for 1D multi-view sequences. 10 seconds of 80 views were compressed by estimating depth maps, encoding only 27 (41 for Flowers) views plus depth maps by modified 3D-HEVC encoder (HTM13) provided by JCT-3V[16], and synthesizing 53 (39 for Flowers) discarded views from decoded data. Depth maps were separately encoded from texture since interactive coding between depth and texture requires long encoding time with small coding gain. When total bit rate of texture and depth is low, this schema yielded about 12 % bit reduction over encoding all 80 views (HTM13). At high bit rate however, synthesis noise degraded average PSNR. Since shorter inter-view distance of reference views improves PSNR of synthesized view but increases total bit rate, optimum inter-view distance exists for each sequence. Applying this coding schema to light-field image coding will improve its coding efficiency.

B. 2D Multi-Vview Image Coding without Depth Map

Even without depth maps, light field image can be efficiently compressed by converting it to multi-view images. Fig.10 bottom shows results of 8×8-view 1920×1080-pixel YUV420 still image Unicorn captured by camera-array, and its compatible light field image of (1920×1080)×(8×8)-pixel [17]. Fig. 10 shows multi-view coding structure (top) and its coding efficiency (bottom). "mv" and "mv-3d" are results of 8×8 multi-view image coding with one-directional inter-view prediction without/with 3D tools. Since maximum number of views for modified HTM13 is 63, 64 views were divided to two 32-view groups. "sv-mf" is a result after converting 8x8-view images to 1-view 64-frame sequence. "lf" is a result of HEVC intra coding of light field image. "lf-mf" is a result after dividing light field image to 8×8 sub-images and then arranging them to 1-view 64-frame sequence for HEVC codec. Multi-view image coding (mv, mv-3d, sv-mf) shows about 100 times higher coding efficiency than light field image coding (lf, lf-mf).

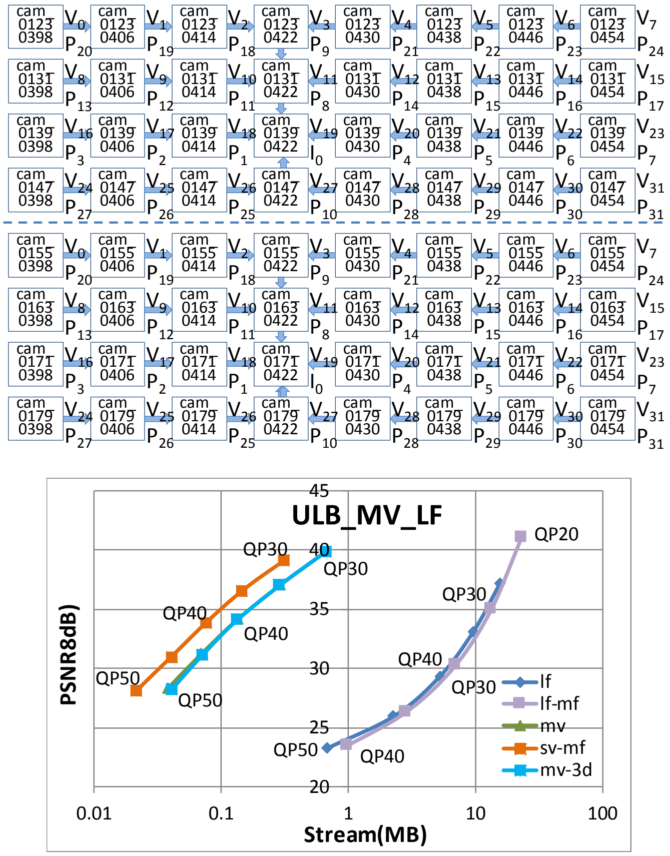
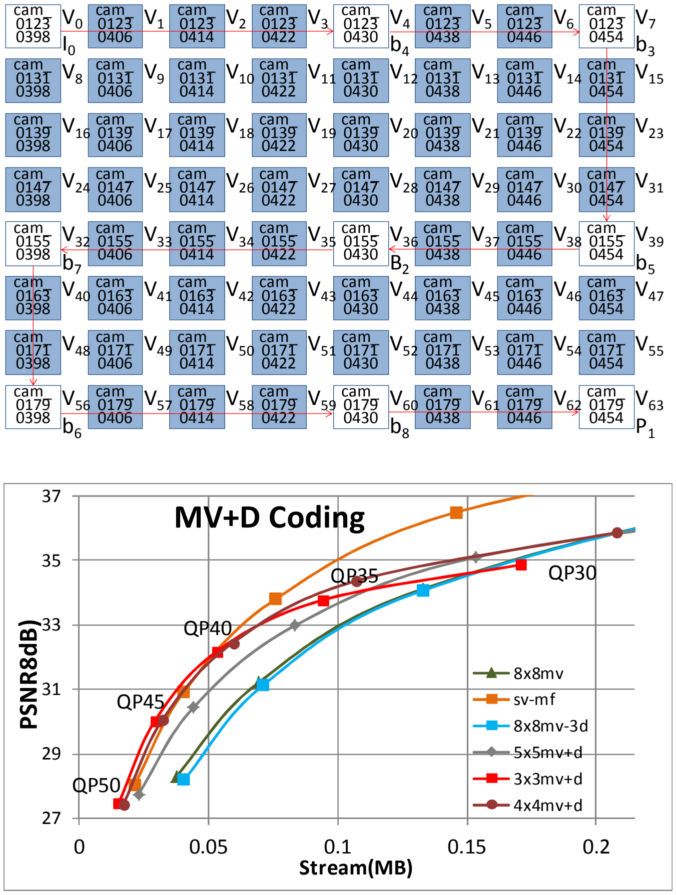


Fig. 10. Multi-view coding structure (left) and coding efficiency (right)[17]

"sv-mf" shows about 2dB higher gain over multi-view coding (mv, mv-3d). This gain seems coming from the translation vector inheritance among views. Rearrangement of light field image to multi-frame sequence (lf-mf) yielded no gain since spatial correlation of light field image is low as seen in Fig. 1.

C. 2D Multi-View Image Coding with Depth Map

We newly encoded same 64-view Unicorn still image by 2D depth estimation at white views shown in Fig. 11 top. Each depth maps were estimated from available up to four neighbor views by the method described in section 3.C. Estimated nine depth maps and associated views were arranged into each single view 9-frame sequences and compressed independently with HEVC codec described in section 5.B. After decoding, 55 dark views were synthesized from nearest four white views and depth maps by the 2D view synthesis method described in section 4.B. Result is in Fig. 11 bottom. When total bit (Stream) is lower than 0.15MB, coding efficiency of 3×3 or 4×4 multi-view plus depth (mv+d) is higher than 8×8 multi-view without depth coding (8×8mv). When total bit is lower than 0.05MB, its coding efficiency is the highest. At large total bit amount, view synthesis noise limited average PSNR of reconstructed 64 views. Also, low correlation among sparse 3×3-views reduced coding gain. In this experiment, since camera parameter (20KB) is negligibly smaller than 9-view images and depth maps (47MB), it was not coded.



cam 0123 0398 V_0 I_0	cam 0123 0406 V_1	cam 0123 0414 V_2	cam 0123 0422 V_3	cam 0123 0430 V_4 b_4	cam 0123 0438 V_5	cam 0123 0446 V_6	cam 0123 0454 V_7 b_3
cam 0131 0398 V_8	cam 0131 0406 V_9	cam 0131 0414 V_{10}	cam 0131 0422 V_{11}	cam 0131 0430 V_{12}	cam 0131 0438 V_{13}	cam 0131 0446 V_{14}	cam 0131 0454 V_{15}
cam 0139 0398 V_{16}	cam 0139 0406 V_{17}	cam 0139 0414 V_{18}	cam 0139 0422 V_{19}	cam 0139 0430 V_{20}	cam 0139 0438 V_{21}	cam 0139 0446 V_{22}	cam 0139 0454 V_{23}
cam 0147 0398 V_{24}	cam 0147 0406 V_{25}	cam 0147 0414 V_{26}	cam 0147 0422 V_{27}	cam 0147 0430 V_{28}	cam 0147 0438 V_{29}	cam 0147 0446 V_{30}	cam 0147 0454 V_{31}
cam 0155 0398 V_{32} b_7	cam 0155 0406 V_{33}	cam 0155 0414 V_{34}	cam 0155 0422 V_{35}	cam 0155 0430 V_{36} B_2	cam 0155 0438 V_{37}	cam 0155 0446 V_{38}	cam 0155 0454 V_{39} b_5
cam 0163 0398 V_{40}	cam 0163 0406 V_{41}	cam 0163 0414 V_{42}	cam 0163 0422 V_{43}	cam 0163 0430 V_{44}	cam 0163 0438 V_{45}	cam 0163 0446 V_{46}	cam 0163 0454 V_{47}
cam 0171 0398 V_{48}	cam 0171 0406 V_{49}	cam 0171 0414 V_{50}	cam 0171 0422 V_{51}	cam 0171 0430 V_{52}	cam 0171 0438 V_{53}	cam 0171 0446 V_{54}	cam 0171 0454 V_{55}
cam 0179 0398 V_{56} b_6	cam 0179 0406 V_{57}	cam 0179 0414 V_{58}	cam 0179 0422 V_{59}	cam 0179 0430 V_{60} b_8	cam 0179 0438 V_{61}	cam 0179 0446 V_{62}	cam 0179 0454 V_{63} P_1

Fig. 11. 2D multi-view image and depth coding schema for 3×3mv+d (top) and codding efficiency (bottom)

VI. Conclusion

An efficient light field image coding method was proposed. It converts light field image to lossless multi-view images, estimates depth maps from up to four reference views, encodes these key views and depth maps by hierarchical bi-directional inter-view prediction coding, and then synthesizes discarded views from decoded views and depth maps. Proposed method was compared to direct light field image coding and multi-view texture only coding. It yielded more than 100 times higher coding efficiency than direct light field image coding and yielded higher coding efficiency than multi-view coding without depth maps at low total bit amount. Since practical light field image consists of huge number of multi-view images of small disparity, it will be compressed largely by this method. Also, since this method enables parallel processing in depth estimation, depth and texture coding, and view synthesis, coding through-put time can be largely reduced. Using a camera-captured light field image and applying a fast depth estimation method to this system is our future work.

References

[1] C. Perra and P. Assuncao, “High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement,” in IEEE International Conference on Multimedia & Expo Workshops (ICMEW) , 2016, pp. 1-4.

[2] R. Verhack, T Sikora, L. Lange, R. Jongebloed, G. V. Wallendae, and P. Lambert, “Steered mixture-of-experts for light field coding, depth estimation, and processing,” in Proc. IEEE International Conference on Multimedia and Expo (ICME), 2017. Pp. 1183-1188.

[3] E.H.. Adelson, and J.R. Bergen, “The Plenoptic Function and the Elements of Early Vision,” In Computation Models of Visual Processing, M. Landy and J.A. Movshon, eds., MIT Press, Cambridge, pp.3-20, 1991.

[4] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “Light-Field vs Multi-View 2,” ISO/IEC JTC1/SC29/WG11, M41029, 119th MPEG Meeting, 2017.

[5] Video, “Report on Experimental Framework for 3D Video Coding,” ISO/IEC JTC1/SC29/WG11, N11631, 94th MPEG Meeting, 2010.

[6] https://opencv.org/.

[7] K. Wegner, M. P. Tehrani, and G. Lafruit, “Description of Exploration Experiments on Free-viewpoint Television (FTV),” ISO/IEC JTC1/ SC29/WG11, N14105, 107th MPEG Meeting, 2013.

[8] G. Lafruit, K. Wegner, and M. Tanimoto, “Call for Evidence on Free-Viewpoint Television: Super-Multiview and Free Navigation,” ISO/IEC JTC1/SC29/WG11, N15348, 112th MPEG Meeting, 2015.

[9] T. Senoh, K. Wakunami, H. Sasaki, R. Oi and K. Yamamoto, “Fast Depth Estimation Using Non-iterative Local Optimization for Super Multi-view Images,” IEEE Global SIP 2015, pp.1042–1046, 2015.

[10] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “Improved Fast DERS,” ISO/IEC JTC1/SC29/WG11, M41028, 119th MPEG Meeting, 2017.

[11] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “Enhanced DERS for Quad Reference Views (eDERS),” ISO/IEC JTC1/SC29/ WG11, M41955, 121th MPEG Meeting, 2018.

[12] http://wg11.sc29.org/, MPEG-Content.

[13] I. Ahn and C. Kim, “Depth-based Disocclusion Filling for Virtual View Synthesis,” IEEE International Conference on Multimedia and Expo, Melbourne, Australia, 2012.

[14] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “VSRS improvement,” ISO/IEC JTC1/SC29/WG11, M38979, 116th MPEG Meeting, 2016.

[15] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “fDEVS Coding Results for Four SMV Test Sequences,” ISO/IEC JTC1/SC29/WG11, M38978, 116th MPEG Meeting, 2016.

[16] http://phenix.it-sudparis.eu/jct3v/.

[17] T. Senoh, K. Yamamoto, N. Tetsutani, and H. Yasuda, “EE Results on Light-Field Compression,” ISO/IEC JTC1/SC29/WG11, M41784, 120th MPEG Meeting, 2017.

ISBN 978-90-827970-1-5 © EURASIP 2018