Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)

Paper

Depth Coding using Downsampling and View based Upsampling

（ダウンサンプリングと画像ベースアップサンプリングを用いる奥行き符号化）

Meindert Onno Wildeboer†, Tomohiro Yendo (member)†, Mehrdad Panahpour Tehrani (member)†, Toshiaki Fujii (member)††, and Masayuki Tanimoto (member)†

Abstract There are several ways to represent 3D scene information. One popular way is based on N-view plus N-depth representation. In applications based on this representation, efficient compression of both view and depth is important. In this paper, we present a depth coding method that uses depth downsampling and a novel depth upsample filter that uses the color view in the depth upsampling. Our method of depth down- and up-sampling is able to reconstruct clear object boundaries in the upsampled depth maps, and, therefore, we can obtain a better coding efficiency. Our experimental results show that the proposed depth resampling filter, used in combination with a standard state-of-the-art video encoder, can increase both the coding efficiency and rendering quality, particularly at lower bit rates.

Key words: depth upsampling, depth coding, depth map, Free-viewpoint television (FTV), 3DTV

1. Introduction

The research and development in 3D video applications such as 3DTV[1] and FTV(Free-viewpoint television)[2)~4)] has increased rapidly in recent years. **Fig. 1** shows a block diagram of an N-view plus N-depth FTV system. A 3D scene is captured by multiple cameras followed by color correction, rectification, and depth estimation, which are done off-line. Then the multiple views and depths are compressed and transmitted or stored. At the receiver-side, the decompressed multiple views and depth maps are used to render novel views using Depth Image Based Rendering[5]. This configuration keeps the required processing at the receiver side low. In applications based on a N-view plus N-depth representation, both view and depth are compressed, and efficient depth compression is important to maintain good rendering quality. In this paper we present a depth down/up-sampling method which is used in combination with a standard video encoder for depth coding. The novelty of our contribution is in the depth upsampling, which we apply to depth coding. The system diagram of our coding system is shown in **Fig. 2**.

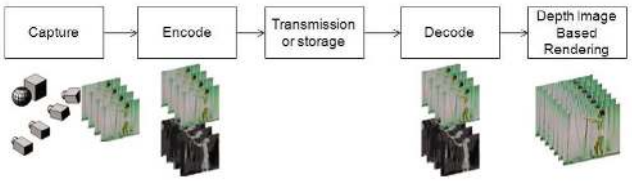


Fig. 1 FTV system based on N-view + N-depth.

A depth map is a gray-scale image representing per-pixel depth relative to the camera. Depth map images tend to contain smooth changing intensity values within objects and sharp changes at object edges, because they represent object surfaces rather than object texture. So depth maps tend to have more spatial redundancy compared to texture video. We use this property in our proposed depth compression method, by downsampling the depth at the encoder.

The approach to downsample the depth before compression has been used before, for example in the works of[8)~10)]. As part of the MPEG 3DV experiments, the authors of[9)] showed that downsampling the depth by a factor of two could improved the coding efficiency on some sequences. In their experiments Nearest Neighbor down- and up-sampling was used. In[10)] a depth reconstruction filter was proposed to correct depth coding errors combined with a down/up-sampling scheme

Received September 6, 2010; Revised January 27, 2011; Accepted February 28, 2011
† Graduate School of Engineering Nagoya University
(Furo-cho, Chikusa-ku, Nagoya, 464-8603, Japan)
†† Graduate School of Science and Engineering, Tokyo Institute of Technology
(2-12-1 Ookayama, Meguro-ku, Tokyo, 152-8550, Japan)

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)

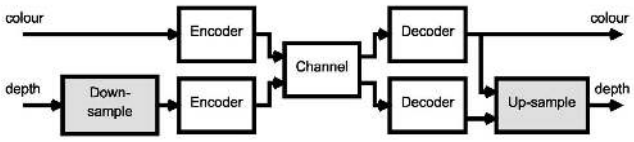


Fig. 2 System diagram of our coding system.

for depth coding. In the works of[8)9)], and[10)], conventional upsampling is used, whereby only neighboring depth samples are used in the upsampling. For example both[9)], and[10)] used nearest neighbor (NN) upsampling. For small scaling factors, NN works relatively well on depth maps because it maintains sharp edges. But NN upsampling can introduced artifacts in the view synthesis result because the edge is not necessarily at the correct position (i.e. not correctly aligned with the color view).

In our proposed scheme, we use the correlation between the color view and the depth map, and therefore we can reconstruct a better upsampled depth map than the conventional resampling methods as used in[8)~10)]. Recently, we proposed a depth coding method using depth upsampling whereby the (decoded) full resolution view is used[11)]. At the same time a very similar approach was proposed by Li[12)]. In both works, a depth value is calculate using weights based on pixel distance and color difference. In our method, a winning weight is selected by winner-takes-all approach, whereas in[12)] a weighted average is calculated, followed by a depth flattening step. But by calculating a weighted average, depth edges are still smoothened which is not the case for our method. This paper contains extended version of our works[11)] including additional experiments and comparison results.

Our paper is organized as follows: We outline our proposed depth down-/up-sampling method in section 2, followed by our experiments and discussion in section 3. We conclude our paper in section 4.

2. Proposed Depth Coding Method

2.1 Depth estimation background

Our proposed depth upsampling method is inspired by techniques from the stereo-matching and depth estimation field. In our method, we downsample depth at the encoder side and therefore receive a lower resolution depth at the decoder. In the decoder we need to upsample the depth so depth of missing samples needs to be estimated. The basic principle of all disparity or depth estimation methods is to find corresponding pixels in two or more images. How well two points match, is generally expressed in a cost measure. One of the simplest cost functions is the absolute difference in luminance of two corresponding pixels. A popular cost function is the adaptive support-weight approach of Yoon and Kweon[13)]. They assign all pixels in a support window a weight, based on their spatial distance and color difference:

$$w(p,q) = \exp\left(-\left(\frac{\Delta c_{pq}}{\lambda_c} + \frac{\Delta g_{pq}}{\lambda_p}\right)\right) \quad (1)$$

where Δg_{pq} and Δc_{pq} represent the Euclidean distance between pixel p and q, and the Euclidean distance between the colors of pixel p and q, respectively. Parameters λ_g and λ_c control the strength of the distance- and color similarity, respectively. As generally neighboring pixel with similar color have similar depth, the cost function of[13)] has proofed to be very effective.

We use a similar philosophy of depth and color similarity in our depth upsampling method. In our case, for depth upsampling we have a sparse set of depth values in the downsampled depth map together with a full resolution color view. We estimate the depth of missing samples based on color similarity and pixel distance.

2.2 Depth downsampling

Depth maps can have large intensity changes at object boundaries which we need to maintain. Most traditional down- and up-sample filters are a combination of a low-pass filter and an interpolation filter. Applying low-pass filters and interpolation filters to depth maps will smooth the objects edges which is undesired. To maintain the object edges, the authors in[10)] proposed downsampling based on a 2D median:

$$Depth_{down}(x,y) = median(W_{s\times s}) \quad (2)$$

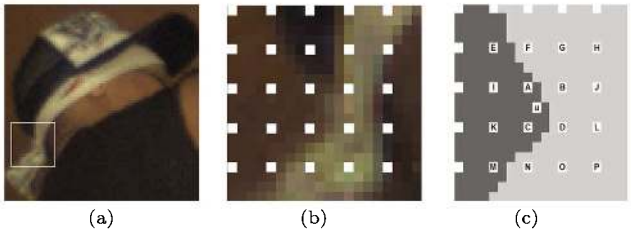
where $W_{s\times s}$ represents an $s\times s$ block in the input depth map, and s is an odd numbered downsampling factor.

As most common video resolutions are a multiple of 4,8, or 16 we mainly consider step sizes of 1,2,4,8 etc., although our method also works for other re-sampling factors. Our depth downsampling is a simple pixel dropping scheme formulated as follows:

$$Depth_{down}(x,y) = Depth_{in}(x*step, y*step) \quad (3)$$

All samples on a regular grid are stored in the low resolution depth map, and pixels off the grid are dropped. **Fig. 3**(a) shows a color image with enlarged sections of the color view and corresponding depth map in (b) and (c), respectively. We overlaid a white sample grid with

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)



(a) (b) (c)

Fig. 3 (a)image section, (b)sample-grid on the color image, (c)sample-grid on the depth map.

step=4 as illustration, and only the depth value of the white pixels are stored in the downsampled depth map. Note that the sample grid overlaid on the color image is for illustration purposes only. The letters in the Fig. 3(c) will be used in section 2.3 for the explanation of our depth upsampling method.

2.3 Depth upsampling

As described in the previous section, the full resolution depth map is downsampled by pixel dropping in the encoder. In the decoder we need to upsample the depth to its original size. Referring to the illustration in Fig. 3, at the decoder we only have depth values for the white pixels. For every pixel u that needs upsampling, we find the four nearest neighbors $N \in \{A, B, C, D\}$ as depicted in Fig. 3. Next we calculate a weight for pixel u and each neighbor based on their pixel distance and color difference as follows:

$$W_N(u) = f(u, N)g(u, N), \tag{4}$$

$$f(u, N) = \|u - N\|, \tag{5}$$

$$g(u, N) = \sum_{C=r,g,b} |C_u - C_N|. \tag{6}$$

where function $f()$ is the L1 pixel distance, and $g()$ the L1 color difference between pixel u and the four neighbors in set $N \in \{A, B, C, D\}$. We could then select the winning depth value by a simple winner-takes-all (WTA) based on the weights W_N, but WTA optimization is sensitive to noise. Therefore we add a smoothing term V_N to form a total energy as follows:

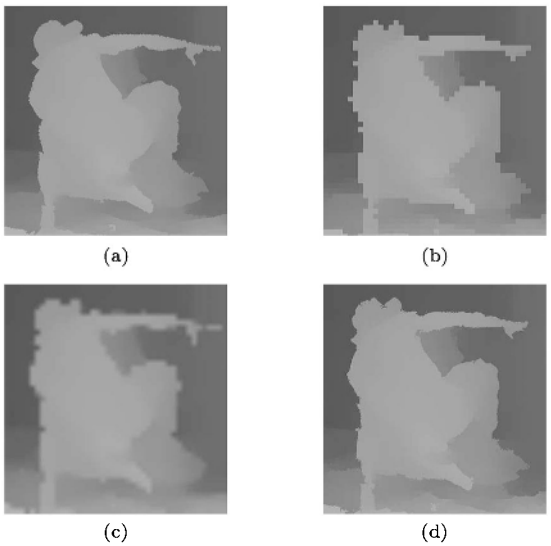
$$E(u) = W_N(u) + V_N, \tag{7}$$

$$V_N = \lambda_1 |d_N - d_L| + \lambda_2 |d_N - d_{up}|. \tag{8}$$

where d_L is the depth of the pixel left of pixel u and d_{up} is the depth of the pixel above u. The parameters λ_1 and λ_2 control the strength of the smoothing. Then we select the winning depth value by WTA:

$$d_u = \text{argmin}(E(u)) \tag{9}$$

In stead of using only four neighbors $N \in \{A, B, C, D\}$, we can easily extend our method to include more neighboring pixels on the sample grid, e.g. the 16 neighbors $N \in \{A, B, ..P\}$ as depicted in Fig. 3.



(a) (b) (c) (d)

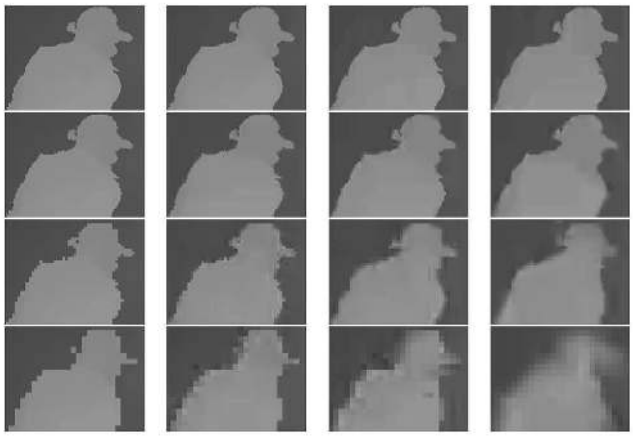
Fig. 4 Upsampling result for step=8, (a) Original depth, (b) Nearest Neighbor, (c) Bilinear, (d) Proposed method.

Even though this method is relatively simple, it is very effective in maintaining sharp edges in the upsampled depth maps. **Fig. 4** shows a section of a depth map which was downsampled by a factor of eight using the described depth downsampling method (section 2.2), and then upsampled using Nearest Neighbor(NN), bilinear, and our proposed upsampling method (with λ_1=λ_2=0, and $N \in \{A, B, C, D\}$). As can be seen in Fig. 4, the Nearest Neighbor method maintains sharp boundaries, but shows strong "blocking", whereas the bilinear algorithm strongly blurs the depth edges. In contrast, our method can recover most sharp depth boundaries. One disadvantage of downsampling based on a regular grid is that small details will be lost, especially for larger step sizes. This effect can be seen in Fig. 4(d), where part of the fingers of the person where lost because the step size of 8 is relatively large in this case.

2.4 Depth coding artifacts

Most state-of-the-art video compression methods such as H.264/AVC[6], and MVC[7] are block based. In brief, the encoder splits the input image into blocks of variable size e.g. 4×4, 8×8, or 16×16 pixels or a combination of these, and performs a 2D DCT (Discrete Cosine Transform). The DCT coefficients are quantized with a given QP (Quantization Parameter), and

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)



(a) Uncoded (b) QP32 (c) QP36 (d) QP40

Fig. 5 Depth coding artifacts. From top to bottom: original resolution, and downsampled by step 2, 4, and 8 respectively.

scanned in zigzag order before entropy coding by VLC (Variable Length Coding) or CABAC (Context-based Adaptive Binary Arithmetic Coding).

To show the type of artifacts introduced by this type of encoder, we compress a depth video with H.264/AVC with QP values of 32, 36, and 40. We run four experiments, with the original (full resolution) depth video, and downsampled versions. The lower resolution depth video is generated by pixel dropping (see section 2.2), with a step of 2, 4, and 8, respectively. **Fig. 5** shows a section of the resulting decoded depth maps. It can be clearly seen from Fig. 5 that the artifacts increase faster for larger downsampling step sizes at a fixed QP. For example, the depth at step=8 as shown in the bottom row of **Fig. 5** degrades much faster from (a)uncoded to (d)QP40, than for the depth at full resolution, or step=2. This is because the smallest encoder block size of 4×4 pixels covers a larger object area in this case.

Furthermore, from the top two rows of Fig. 5 we note that the decoded depth maps contain small coding errors at the object edges. The area of these errors are limited to only one or a few pixels around the object boundary. These small errors can be efficiently corrected with a 2D Median filter. Therefore we apply a 3×3 Median filter to the decoded depth maps, before upsampling. As the results in Fig. 5 show, for larger downsampling step sizes and increasing QP values, the area of the coding error increases to more than a few pixels. For this reason, we only apply the filter for downsampled depth of step size 1 and 2 in our experiments.

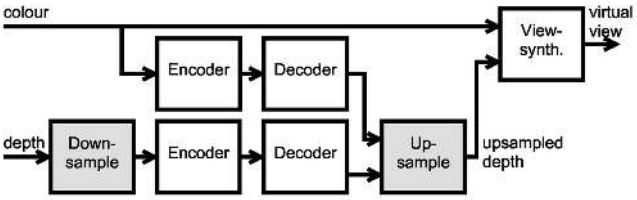


Fig. 6 System setup for our experiments.

3. Experiments and Analysis

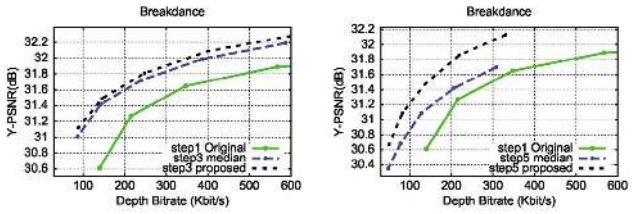
We analyze the performance of our method on 32 frames of six sequences, namely the Microsoft MVC sequence[14] "Breakdance", and the MPEG test sequences [15] "Champagne-tower", "Kendo", and "Balloons" from Nagoya University, Japan, "Newspaper" from GIST, Korea, and "Book arrival" from HHI, Germany. To measure the quality of the depth, we render a novel view using the decoded depth and color views. For all sequences we select three cameras and generate a virtual view at the position of the center camera. For example, for Champagne-tower we use cameras 39 and 41 as input and generate a virtual view at the position of camera 40. We use the MPEG View Synthesis Reference Software (VSRS3.5)[16] for generating the rendered views. VSRS takes two views and two depth maps as input. Therefore we compress both the left and right depth videos and calculate the total depth bit rate. We encode the depth maps using H.264/AVC with an I-B-P coding structure, a GOP-size of 0.5 seconds, and CABAC (Context Adaptive Binary Arithmetic Coder) as entropy mode.

For compressing the depth maps of original resolution (we refer to these as step=1) we use QP 32,36,40,44, and 48. For step=2,4, and 8, we first downsample the left and right depth maps and then compress them with QP 24,28,32,36, and 40. After decoding, the depth maps are upsampled and used for view synthesis. In all experiments we use a smoothing factor of $\lambda_1=\lambda_2=0.2$ for our upsampling method.

As objective measurement, we calculate the average Peak Signal to Noise Ratio (PSNR) between the rendered virtual view and the original camera.

The system setup as used in our experiments is illustrated in **Fig. 6**. As the color view is used in our upsampling method, we also compress the color view as would be the case in a practical application. For all experiments we compress the view with QP22. Note that we compress depth and view as described above for all experiments, except for the experiments in section 3.2

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)



(a) (b)

Fig. 7 Coding performance of method[10] versus our proposed method (a) step=3, (b) step=5.

where no compression was used.

3.1 Comparison with conventional method[10]

As mentioned in section 2.2, in[10] a median based downsampling scheme was used. As upsampling method,[10] uses NN interpolation defined as:

$$Depth_{up} = Depth_{down}\left(\left\lfloor \frac{x}{step} \right\rfloor, \left\lfloor \frac{y}{step} \right\rfloor\right) \quad (10)$$

followed by a 2D median filter. We compare the coding performance of the complete coding method of[10] with our proposed method on the Breakdance sequence, for a step size of 3 and 5. **Fig. 7** shows the PSNR of the synthesized view versus the depth bit rate. As can be seen from Fig. 7, at step=3 our method performs marginally better (max 0.1dB). The advantage increases for larger step sizes such as step=5(up to 0.4dB). These results shows the benefit of using the color view in upsampling, compared with conventional upsampling using only neighboring depth values.

3.2 Depth upsampling and step size

Next we investigate the influence of the scaling factor (step size), and compare our method with NN upsampling, and the method of Li[12] for different step-sizes. **Fig. 8** shows the PSNR of the synthesized view for step sizes of 1, 2, 4, and 8. We used no coding, but perform depth down- and upsampling, and view synthesis only. It is obvious from the plots in Fig. 8, that our proposed method performs much better than NN at higher step sizes, showing the benefit of using the color view. At the smallest step size of 2, our method performs similar to NN. Only in case of Champagne-tower it is slightly worse (0.04dB), but on the other sequences it is 0.04dB higher on average.

As mentioned, our method is similar to the works of [12]. The main difference is that[12] calculates a weighted average in their first step, rather than choosing the winning depth. This results in generally smoother depth maps, whereby also the depth edges become slightly

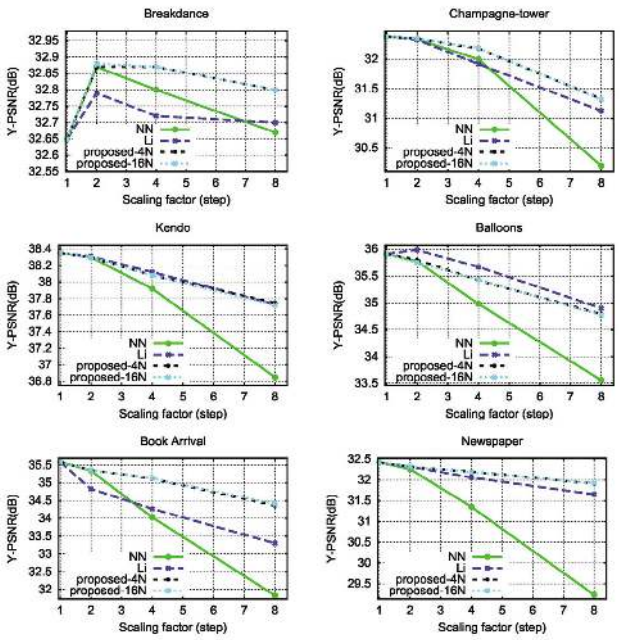


Fig. 8 View synthesis quality of proposed depth upsampling method (with N is 4 and 16 neighbors) versus Nearest Neighbor (NN) and Li[12] for step=1,2,4,and 8.

blurred. In a second step, the depth flattening step, small depth jumps are flattened to create a predominantly flat depth map[12]. The slight blurring of object edges does not occur in our method, as we do not perform averaging but chose the most likely neighboring depth value. As can be noted from Fig.8, our method performs better than[12] on four sequences, about same on Kendo, slightly worse on Balloons. If the depth maps contain clear object edges in line with the color view, our method tends to perform better, because it does not blur object edges. On the other hand, in some cases smoother depth can even slightly improve the view-synthesis, as for example for the Balloons sequence. The Balloons sequence has many objects at different depths, and some of the depth edges are accurate, some are misaligned. Here the result of[12] seems beneficial. Breakdance generally has clear depth edges, but is the only non-rectified sequence in our test, which causes shift and rotation in the synthesized view. The resampled depth obtains higher PSNR than the original depth. The slightly blurred object edges of[12] reduces its performance even a bit below NN upsampling.

Furthermore, we compared our method using 4 neighboring pixels in the upsampling ($N \in \{A, B, C, D\}$, we denote this as 4N), and 16 neighboring pixels ($N \in \{A, B, C..P\}$, denoted as 16N) in Fig. 8. These experiments show that both 16N and 4N performed very

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)

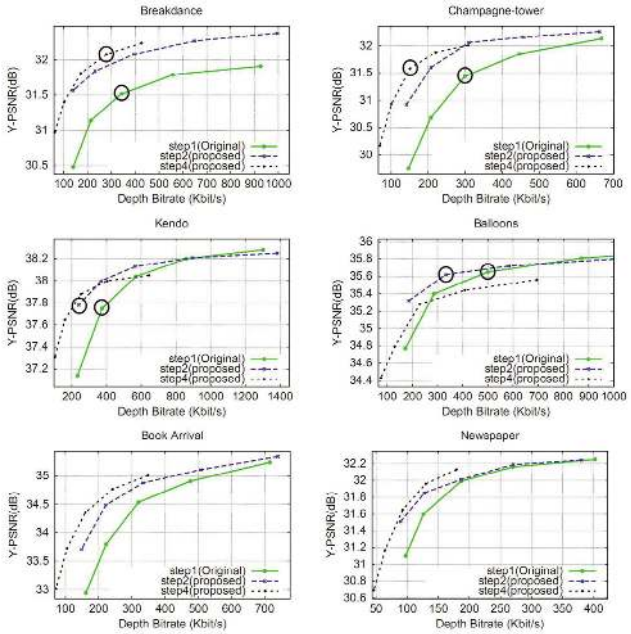


Fig. 9 PSNR of the synthesized view versus total depth bit-rate; View synthesis results for the marked rate points are shown in **Fig. 10-13**.

similar. Considering that 16N is more computational intensive than 4N, the additional performance seems not worth the additional computational cost. Therefore, we use 4N only in our further experiments.

3.3 Depth down-/up-sampling and compression

In the experiments in this section we want to find out if our proposed down- and up-sampling method combined with a state-of-the-art encoder such as H.264/AVC can improve the coding efficiency. The goal is to find out if it is beneficial to use downsampled depth and compress it less hard using H.264/AVC, or if it is better to compress the full resolution depth stronger. **Fig. 9** shows the Rate-Distortion plots for all our test sequences, where the PSNR of the synthesized virtual view is plotted versus the total depth bit rate. In our experiments we included step sizes of 1,2,4, and 8, but a step size of 8 showed only better results at very low bit rates. At those bit rates the visual quality of the virtual view decreases fast, so to improve the readability of our plots we did not include step 8.

The plots in Fig. 9 indicate that our proposed method improves the coding efficiency in terms of PSNR for all test sequences, especially for lower bit rates. To evaluate the visual performance, we selected two rate points for 4 of the test sequences, namely Champagne-tower, Kendo, Breakdance and Balloons. We selected one rate point with the depth at original resolution (step=1) and a rate point with similar PSNR but lower bit rate of our proposed method, for Champagne-tower, Kendo and Balloons. For Breakdance we selected two points around 300kbit/second to show the quality improvement at similar bit rate. The selected rate points have been marked, in Fig. 9, and the PSNR and total depth bit rates are summarized in Table 1. A snapshot of the corresponding synthesized virtual views can be found in **Fig. 10** to **Fig. 13**, respectively. From Fig. 10-13(a) and (b) we can see that with our method we can obtain similar quality synthesized views at lower bit rates. We have also enlarged two sections in Fig. 10-13(c)-(e) to show areas where the view quality using our method is even slightly better.

Table 1 Selected rate points for **Fig. 10-13**.

	PSNR (dB)		Depth bit rate (kbit/s)	
Sequence	Original	Proposed	Original	Proposed
Champagne-tower	31.4	31.6	299	150
Kendo	37.8	37.9	375	252
Breakdance	31.5	32.1	343	278
Balloons	35.7	35.6	500	332

3.4 Analysis

Generally, using the (uncompressed) original depth gives better view synthesis quality than using (uncompressed) down-/up-sampled depth, as can be seen from Fig.8. Our upsampling method is based on the assumption that pixels with similar color have similar depth, and object edges align with texture edges. This is generally true for many scenes and for ground truth depth, but unfortunately for (practical) depth maps, estimated by depth estimation methods, this assumption is often violated. In all our experiments, we used estimated depth maps as no ground truth is available, and some depth edges show "oversmoothing", or pixels with same color do not have same depth. These aspects make the performance analysis somewhat complicated.

Our method is mainly beneficial for lower bit rates, where strong compression of the full resolution depth map causes blurring and other coding artifacts in the decoded depth maps. By downsampling the depth, the spatial redundancy is reduced, and less compression is required to obtain the same bit-rate as using the original depth maps. By using the color view in the upsampling, our method can restore clearer object edges in the depth maps. This results in increased coding efficiency for lower bit rates, as was shown in our experiments (see

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)

Fig.9).

Furthermore, note that for example for a step size of 4, the depth edge can be shifted only 3 pixels under influence of the color view, so only slight misaligned depth/color edges can be compensated by the down-/up-sampling. Therefore, the coding performance is different for each sequence, depending on the scene and quality of depth etc.

4. Conclusion

In this paper we have proposed a depth coding method containing depth downsampling and a novel depth upsampling method. The lower resolution decoded depth map and the corresponding high-resolution color view are used to calculate a weight-cost based on color similarity and distance. During upsampling, the depth of missing pixels is estimated based on the weight-cost. Our experiment results showed that making use of the color view in depth upsampling is beneficial over conventional upsampling methods. We have also shown that the proposed method can increase the depth coding efficiency compared with coding the full resolution depth. We can improve coding efficiency mainly for lower depth bit rates, where compression of the full resolution depth causes blurring artifacts.

But our method also has some limitations, namely, for large re-sampling step sizes, small details or small objects in the depth maps cannot be restored due to our regular downsampling grid. As future research, we are planning to investigate asymmetrical downsampling methods to further improve our method.

This research is partially supported by Strategic Information and Communications R&D Promotion Programme (SCOPE)093106002 of the Ministry of Internal Affairs and Communications. We would like to thank Microsoft Research for providing the "Breakdance" sequence and depth maps.

〔References〕

1) A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, "multiview imaging and 3DTV", Signal Processing Magazine, 24(6):10-21 (2007)
2) T. Fujii, "A Basic Study on Integrated 3-D Visual Communication", Ph.D dissertation in engineering. The University of Tokyo (1994)
3) T. Fujii, M. Tanimoto, "Free-viewpoint Television based on the Ray-Space representation" Proc. SPIE ITCom,**4864-22**,175-189 (2002)
4) M. Tanimoto, "Overview of free viewpoint television", Signal Process.: Image Communication, **21**,6, pp. 454-461(Jul.2006)
5) C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3D-TV", Proc. SPIE Stereoscopic Displays and Virtual Reality Systems, XI, pp.93-104 (Jan.2004)
6) ITU-T, ISO/IEC FCD 14496-10, "Advanced video coding for generic audiovisual services," ITU Recommendation H.264 and ISO/IEC 14496-10 MPEG4-Part 10:AVC
7) ISO/IEC JTC1/SC29/WG11, "Text of ISO/IEC 14496-10: 200X/FDAM 1 Multiview Video Coding," Doc. N9978, Hannover, Germany (Jul. 2008)
8) S. Shimizu, M. Kitahara, H. Kimata, K. Kamikura, and Y. Yashima, "View scalable multiview video coding using 3-D warping with depth map," IEEE Transactions on Circuits and Systems for Video Technology, **17**, pp. 1485・495, (Nov.2007)
9) ISO/IEC JTC1/SC29/WG11, "Influence of views and depth compression onto quality of synthesized views," M16758, London, UK (Jul. 2009)
10) K-J. Oh, S. Yea, A. Vetro, Y-S. Ho, "Depth Reconstruction Filter and Down/Up Sampling for Depth Coding in 3-D Video", IEEE signal processing letters, **16**,9, p747-750 (Sep.2009)
11) M.O. Wildeboer, T. Yendo, M. Panahpour Tehrani, T. Fujii, M.Tanimoto, "Color Based Depth Up-sampling for Depth Compression", Picture Coding Symposium (PCS), pp. 170-173, Nagoya (Dec.2010)
12) Y. Li, L. Sun, "A Novel Upsampling Scheme for Depth Map Compression in 3DTV System," Picture Coding Symposium (PCS), pp. 186-189, Nagoya (Dec.2010)
13) K.-J. Yoon and I.-S. Kweon, "Adaptive Support-Weight Approach for Correspondence Search", IEEE Trans. Pattern Analysis and Machine Intelligence, **28**, pp. 650-656 (2006).
14) C.L. Zitnick, S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," ACM SIGGRAPH, ACM Trans.Graph, **23**,3, pp. 600-608 (Aug.2004)
15) ISO/IEC JTC1/SC29/WG11: "Coding of moving pictures and audio, N9783", Archamps, France (2008)
16) ISO/IEC JTC1/SC29/WG11: "View Synthesis Algorithm in View Synthesis Reference Software 2.0 (VSRS2.0)," MPEG Doc. M16090, Lausanne, Switzerland (2009)

Meindert Wildeboer received the MSc degree in Electronics from the University of Hertfordshire, UK, in 1996. He worked in R&D of several companies before joining the Graduate School of Engineering, Nagoya University, where he is currently working as a researcher. His current research interests include 3D image processing and computer vision.

Tomohiro Yendo received the B.E., M.E., and Ph.D. degrees from Tokyo Institute of Technology, Tokyo, Japan, in 1996, 1998, and 2001, respectively. He is now an assistant professor at the Graduate School of Engineering, Nagoya University. His current research interests include 3-D image display, capturing and processing.

Mehrdad Panahpour Tehrani received Dr.E. degree in Information Electronics from Nagoya University, Nagoya, Japan in 2004. Currently, he is working as an Associate Professor at the Graduate School of Engineering, Nagoya University, Japan. His research interests are 3D image processing and communication.

Toshiaki Fujii received the Dr.E. degree in Electrical Engineering from the University of Tokyo in 1995. He is currently an Associate Professor in the Graduate School of Science and Engineering, Tokyo Institute of Technology. His current research interests include multi-dimensional signal processing and their applications for ITS.

Masayuki Tanimoto received the B.E., M.E., and Dr.E. degrees in electronic engineering from the University of Tokyo, Tokyo, Japan, in 1970, 1972, and 1976, respectively. Since 1991, he has been a Professor at Graduate School of Engineering, Nagoya University. His current research interests include FTV and ITS. Dr. Tanimoto is a Fellow of ITE.

Information and Media Technologies 6(4) : 1221–1228 (2011)
reprinted from: The Journal of The Institute of Image Information and Television Engineers 65(4) : 497–504 (2011)

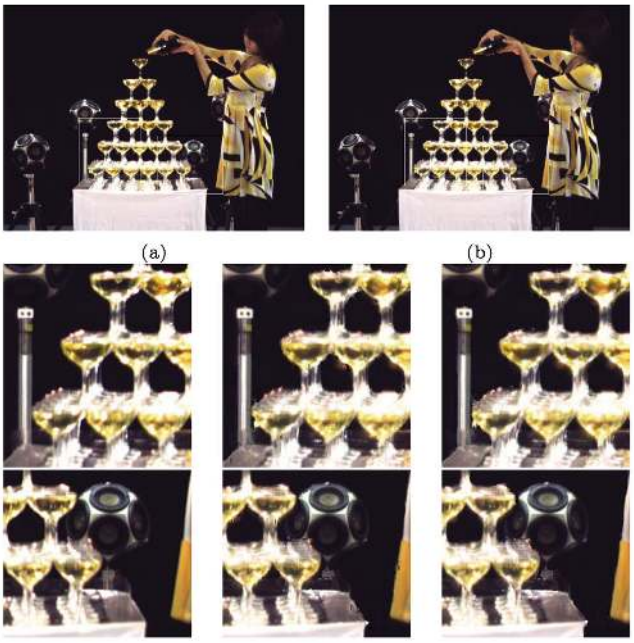


Fig. 10 View synthesis results Champagne-tower: (a) step1 (299kbps/31.4dB), (b) step4 (150kbps/31.6dB), and enlarged sections, (c) ground truth, (d) step1 (original) (e)step4 (proposed).

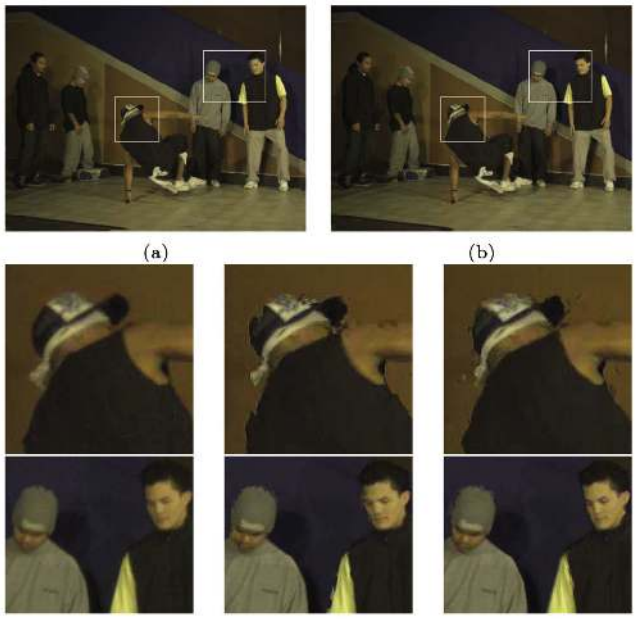


Fig. 12 View synthesis results Breakdance: (a) step1 (343kbps/31.5dB), (b) step4 (278kbps/32.1dB), and enlarged sections, (c) ground truth, (d) step1 (original) (e)step4 (proposed).

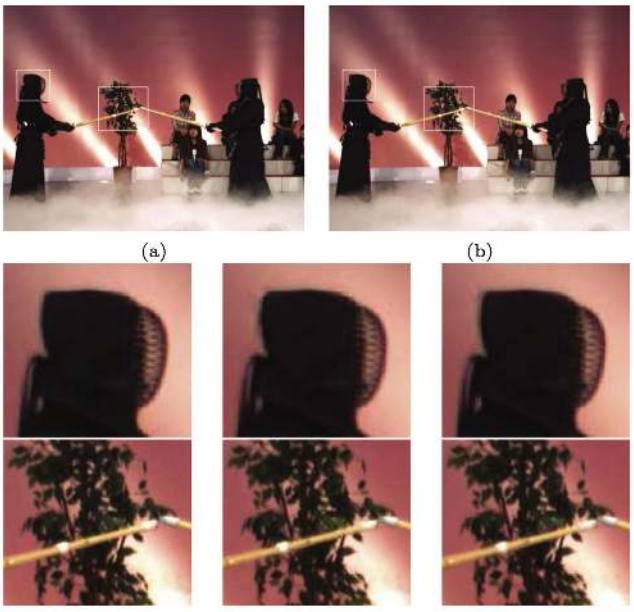


Fig. 11 View synthesis results Kendo: (a) step1 (375kbps/37.8dB), (b) step4 (252kbps/37.9dB), and enlarged sections, (c) ground truth, (d) step1 (original) (e)step4 (proposed).

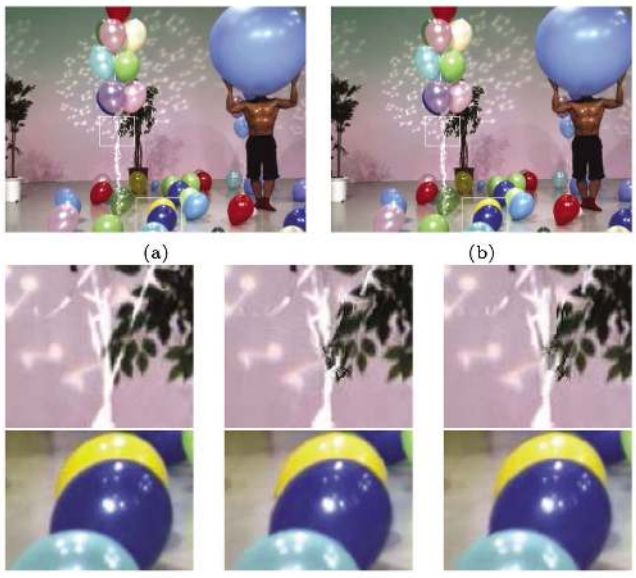


Fig. 13 View synthesis results Balloons: (a) step1 (500kbps/35.7dB), (b) step2 (332kbps/35.6dB), and enlarged sections, (c) ground truth, (d) step1 (original) (e)step2 (proposed).