Progressive Compression of Surface Light Fields

Masaki Kitahara†*, Hideaki Kimata, Kazuto Kamikura, and Yoshiyuki Yashima*

Abstract

A surface light field is a light field parameterized to the surface of a geometric model. Although it enables the generation of high-quality images from arbitrary viewpoints, a huge amount of data is required. Therefore, data compression is needed for efficient transmission and storage. This paper describes a data compression scheme that enables both progressive compression and memory-efficient rendering and presents experimental results.

1. Introduction

Interactive three-dimensional (3-D) content is expected to be used in next-generation visual content applications on the Internet and other online communication applications [1]. It allows the user to freely change views and interactively modify the properties of a scene. These features are suitable for many kinds of applications, ranging from virtual shopping malls that display homes, furniture, and other goods to virtual museums that can be used for educational purposes. One of the key technologies that will enable the creation of such 3-D content is image-based rendering (IBR) [2].

In the IBR framework, multi-view images of a scene or an object are given as input and images taken from different views are generated by interpolating the data obtained from the input images. The surface light field (SLF) method [3] in the IBR framework uses the geometry of the object to render high-quality images of the object from arbitrary viewpoints. A light field can be defined as a function that maps the position and direction to radiance source [4]. In the SLF method, the light field is parameterized on the surface of the geometric model, and the SLF is given as resampled light field data at the sampling points in the parameter space. Because the amount of the SLF data is huge, data compression is needed for efficient transmission and storage.

One highly important function required for data compression algorithms is scalability. Although scalability can be interpreted in many ways, in this paper, we consider signal-to-noise ratio (SNR) scalability. Hereafter, we will refer to the SNR scalable data compression scheme as a "progressive compression" scheme. The texture-based coding proposed by Magnor *et al*. in [5] can be used for progressive compression of the SLF. Although progressive compression is possible by their method, as reported in [5], they reconstruct the SLF with respect to every ray, prior to rendering. This requires a large amount of runtime memory during rendering, and the data size depends on the SLF resolution. To enable memory-efficient rendering, only the data needed for the current view should be reconstructed "on the fly" during rendering. Magnor's method does not provide such functions.

To achieve on-the-fly reconstruction, in our progressive compression scheme we used a compressed representation of the SLF given by a view basis and surface coefficients, which is introduced in section 2. Wood *et al*. showed that this compressed representation allows on-the-fly reconstruction while maintaining fast rendering [6]. Section 3 describes a data compression model based on this compressed representation. Our scheme enables both progressive compres-

† NTT Cyber Space Laboratories
Yokosuka-shi, 239-0847 Japan
E-mail: kitahara.masaki@lab.ntt.co.jp

sion and on-the-fly reconstruction from the compressed representation. While the scheme itself does not address any particular implementation, many are possible. Section 4 shows an example implementation and evaluates its performance and characteristics.

2. Compressed representation using a view basis and surface coefficients

Denoting u as an arbitrary 3-D point on the surface of the geometric model and w as an arbitrary direction vector, we can define a function $L(u, w)$ that maps (u, w) to a color (for example, red, green, and blue intensity) of the light reflected at point u on the object in direction w (**Fig. 1**). The SLF is a light field obtained by discretely sampling $L(u, w)$ with respect to (u, w). We denote SLF $L(u_i, w_j)$ as $L_{i,j}$, where u_i ($i = 0, 1, .., I - 1$) and w_j ($j = 0, 1, .., J - 1$) are both discrete sampling points of u and w. For simplicity, we use $L_{i,j}$ to represent one component of the color. We consider the approximation of the SLF in the following form.

$$L_{i,j} \approx \sum_{k=0}^{K-1} c_{i,k} b_{j,k} \qquad (1)$$

We call the sequence $b_{j,k}$ ($j = 0, 1, .., J - 1$) the "view basis" vector and $c_{i,k}$ ($i = 0, 1, .., I - 1$) the "surface coefficients". We denote the axis of the coordinate system, where directions w_j are defined, as $\mathbf{A}_l$ ($l = 0, 1, 2$). It is known that the following transformation of the coordinate system at each point u_i using the normal $\mathbf{n}_i$ at that point can increase the coherence of the SLF $L_{i,j}$ for the same direction index j [6]. Although the formula in [6] is different from the formula presented above, the objectives of these formulas are the same, and the formula in [6] can easily be applied to the scheme presented in the next section.

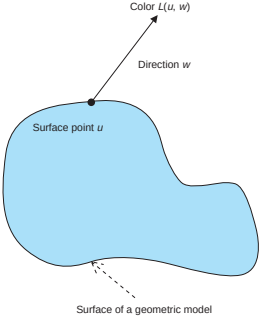

Fig. 1. Surface light field function.

$$\mathbf{A}_{l,i} = 2(\mathbf{n}_i^{\mathrm{T}}\mathbf{A}_l)\mathbf{n}_i - \mathbf{A}_l \qquad (2)$$

This enables a good approximation for smaller values of K (number of basis vectors).

In their method, Wood *et al*. evaluated the median value $\overline{L}_i$ of the data at each point u_i on the geometric model (they called these median values "diffuse texture") and derived the view basis and surface coefficients by their method principal function analysis (PFA). This can be interpreted as having $c_{i,0}$ ($i = 0, 1, .., I - 1$) as $\overline{L}_i$, and $b_{j,0} = 1$ for all j, and the rest of the view basis vectors and the surface coefficients can be derived by PFA. Incorporating the diffuse texture into representation (1) ensures the recovery of the diffuse component-like (view-independent) textures.

3. Our scheme

In our scheme, the encoded view basis and the progressively encoded surface coefficients are placed in the bitstream progressively. In particular, the encoded data of the diffuse texture is placed at the beginning of the bitstream. After that, the encoded data for the pair of view basis vector and surface coefficient set for the same value of k (i.e., view basis vector $b_{j,k}$ ($j = 0, 1, .., J - 1$) and the surface coefficient set $c_{i,k}$ ($i = 0, 1, .., I - 1$) are placed in the order of importance. Here, "importance" means smaller total squared reconstruction error when the pair is added to the approximation in equation (1). The diffuse texture, each view basis, and each surface coefficient set (both with respect to k) are encoded independently, so that they can be independently decoded. For the encoding of the diffuse texture and the surface coefficients, they are wavelet transformed and progressively encoded to obtain a finer-grained bitstream (i.e., in order to refine the progressively reconstructed SLF, fewer bits need to be decoded). On the other hand, the data size of the uncompressed view basis is very small compared with the surface coefficients and can be regarded as side information for recovering the SLF by equation (1). Hence, they are quantized and encoded by a fixed length code and placed right

before the encoded data of the surface coefficients in the bitstream.

The decoding and rendering can be done in the following way. The diffuse texture, the view basis, and the surface coefficients are decoded offline prior to rendering. During rendering, images from novel views are generated directly from the compressed representation of equation (1), where only the data needed for the current view is reconstructed (on-the-fly reconstruction). For example, the algorithm presented in [6] can be used for rendering. In this way, complete reconstruction for all surface points u_i and directions w_j can be avoided, thus reducing the required runtime memory.

Under the compressed representation of equation (1), the optimization techniques for principal components analysis (PCA) searches for sets of optimum view basis and surface coefficients in the mean squared error (MSE) sense, so it is a good candidate for the optimization. Since the values for the SLF $L_{i,j}$ cannot be defined for ones in which the direction of the light points inward from the geometric model, an optimization method that can cope with the missing data must be used. Such PCA-based methods have been proposed in past research, and they can be used for the optimization in our compression model. Furthermore, the PFA proposed by Wood *et al*. can also be used.

Wavelet transform is independently applied to the diffuse texture $\overline{L}_i$ ($i = 0, 1, .., I-1$) and to the surface coefficient set for a fixed value of k (i.e., $c_{i,k}$ ($i = 0, 1, .., I-1$)). For wavelet transform, either "first-generation wavelets (FG wavelets)" [7] or "second-generation wavelets (SG wavelets)" [8] can be used. In either case, the surface sampling points u_i ($i = 0, 1, .., I-1$) must be generated in a way in which the wavelet transform is applicable (this must be done prior to the resampling of the SLF). For the FG wavelets, the surface of the geometric model must be parameterized to a rectangular 2D plane, and the surface must be regularly sampled with respect to the axis of the 2D plane to obtain u_i. For instance, this can be done using the method proposed by Magnor *et al*. [5]. On the other hand, to use SG wavelets, we must generate surface sampling points u_i ($i = 0, 1, .., I-1$) with subdivision connectivity [9]. This can be done by methods such as MAPS [9]. In the following, we denote the wavelet transform coefficients of the diffuse texture and surface coefficients $c_{i,k}$ ($i = 0, 1, .., I-1$ and $k = 0, 1, .., K-1$) as $c'_{i,k}$.

The encoding of the wavelet transform coefficients is processed independently for the diffuse texture $\overline{L}_i$ ($i = 0, 1, .., I-1$) and the surface coefficient set of fixed k (i.e., $c_{i,k}$ ($i = 0, 1, .., I-1$)). For FG wavelets, encoding of the wavelet transform coefficients is the same as in 2D image compression, and progressive compression algorithms such as SPIHT [10] can be used. On the other hand, for SG wavelets, a modified version of SPIHT such as [11], can be used.

In most cases, SLF consists of color data usually with three color components. In these cases, i) the derivation of the view basis and the surface coefficients, ii) the wavelet transform, and iii) the encoding of the diffuse texture, view basis, and the surface coefficients are carried out independently for each color component, and the encoded data is integrated into one bitstream. In particular, encoded data of the view basis vector of a particular k for each color component is placed in an arbitrary order agreed upon by both the encoder and the decoder before the corresponding encoded data of the surface coefficients. The encoded data of the surface coefficients for each color component is interleaved to preserve bitstream progressiveness. **Figure 2** explains this bitstream

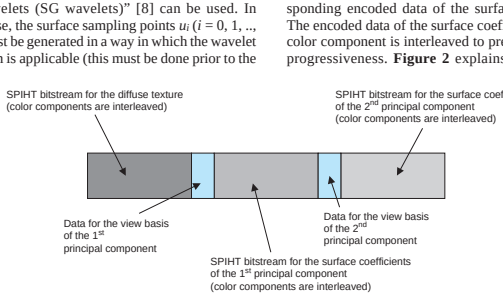

Fig. 2. Bitstream structure.

structure.

4. Experimental implementation of the proposed scheme

4.1 Example implementation

The scheme presented in the last section provides a framework for enabling the functionality explained in section 1. Various implementations are possible, depending on the wavelet filters used, the optimization method used to derive the view basis and the surface coefficients, and so on. In this section, we present an example implementation of our scheme and provide some experimental results.

Multi-viewpoint images (with corresponding camera parameters given) and a geometric model were given to generate the SLF. The geometric model was a triangular mesh. Here, we describe the experiments for two datasets: "horse 1" and "horse 2". The horse 1 dataset consists of 110 images of a real-world object and a mesh that approximates the object's surface. The horse 2 dataset consists of 281 rendered images, generated by texture mapping a texture to the mesh of the horse 1 dataset.

The surface sampling points u_i ($i = 0, 1, .., I - 1$) were generated by the MAPS [9] parameterization algorithm to use the SG wavelets. In our experiments, 45,954 sampling points were generated ($I = 45954$). For the SLF directions, a geodesic dome consisting of 258 vertices was generated in the global coordinate system by subdividing an octahedron, where w_j ($j = 0, 1, .., J - 1$) are given as directions that point outward from the center of the dome to the vertices of the dome ($J = 258$). The SLF was resampled with respect to the local coordinate system $\mathbf{A}_{l,i}$ ($l = 0, 1, 2$) defined by equation (2) at each sampling point u_i. To obtain the resampled SLF, the color at the point in each image that u_i was projected to was evaluated by bilinear interpolation. This was done only for images where u_i was visible. This resulted in a set of color data, with corresponding directions at each point u_i. A method similar to the nearest neighbor sampling was applied to obtain color for each direction w_j. For each direction w_j at each point u_i, the color that was closest in direction (i.e., the color with the smallest inner product of the direction vector) was mapped.

The diffuse texture $\bar{L}_i$ was computed as the mean of the data at each surface point u_i. Denoting ϑ_i as the set of direction indices j, where w_j points outwards from the mesh, and $|\vartheta_i|$ as the volume of the set ϑ_i, the diffuse texture is given by

$$\bar{L}_i = \sum_{j \in \vartheta_i} \frac{L_{i,j}}{|\vartheta_i|}. \tag{3}$$

We define the residual $L'_{i,j}$ of the SLF with respect to $\bar{L}_i$ by

$$L'_{i,j} = L_{i,j} - \bar{L}_i. \tag{4}$$

The view basis and the surface coefficients were derived for the residual L'_i by the NIPALS algorithm, which is an optimization algorithm for PCA that can cope with missing values. There are several versions of this algorithm that can cope with missing values. We used the version proposed by Christofferson, which is explained in a paper by Grung and Manne [12].

The filter used for the wavelet transform was the "modified butterfly" filter [13]. For encoding of the wavelet transform coefficients, the modified SPIHT [11] was used. The entropy coding of the significance values [10] was not used in our implementation. For SPIHT, the input data must be fixed-point binary data that can be treated as an integer. In this example, the wavelet coefficients were rounded to integer values and fed into the SPIHT encoder. On the other hand, the view basis vectors were scalar quantized with 8 bits (because the view basis vectors were normalized, each element in the vectors was bounded by –1 and 1) and encoded with a fixed-length code.

4.2 Experiments

We defined the PSNR (peak signal to noise ratio) for the SLF $PSNR_{SLF}$ by

$$PSNR_{SLF} = 10 \log_{10}(255^2/MSE_{SLF}). \tag{5}$$

The MSE_{SLF} in the above equation is given by

$$MSE_{SLF} = \frac{\sum_{i=0}^{I-1} \sum_{j \in \vartheta_i} \left(L_{i,j} - \bar{L}_{i,j}\right)^2}{\sum_{i=0}^{I-1} \sum_{j \in \vartheta_i} |\vartheta_i|}. \tag{6}$$

The $\bar{L}_{i,j}$ in the above equation was the reconstructed SLF obtained by equation (1).

First, we present the rate-distortion (RD) results for the horse 1 and horse 2 data set in **Figs. 3** and **4**, where the SPIHT encoding was carried out losslessly. The results presented are for the R signal of the color components. The plots ■, ▲, ● are all results

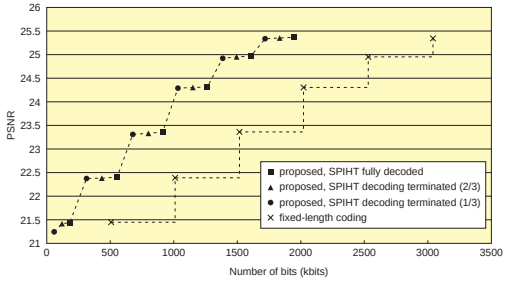

Fig. 3. RD performance for horse 1.

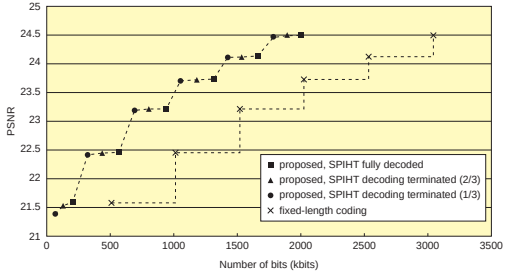

Fig. 4. RD performance for horse 2.

for the proposed scheme, and all the plots were obtained from the same bitstream (i.e., obtained by progressive reconstruction). The plots ■ point to the RD performance when either the diffuse texture or a pair of the view basis vector and the corresponding surface coefficient set was completely decoded. The plots ▲, ● point to the RD performance when the decoding of the SPIHT data for the diffuse texture or the surface coefficient set was terminated at 1/3, 2/3 of the corresponding SPIHT data. Plots × are results for the method we present for comparison. The diffuse texture, the view basis, and the surface coefficients were all encoded by fixed length coding. To be specific, the view basis was encoded in the same way as in the proposed scheme, while the diffuse texture and the surface coefficients were encoded by a fixed-length code with the smallest number of bits needed to represent all the rounded integer values.

The plots ■ indicate that overall the rate of increase of the RD performance was roughly higher when the rate was low, indicating the RD characteristics that progressive compression schemes should have. The reason for this is that the more important the pair of view basis vector and surface coefficient set was, the closer that pair was to the beginning of the bitstream. Comparing the RD performance with the × plots,

there was a significant compression gain. More importantly, the plots ● indicate that the decoding of the surface coefficients can be terminated and still increase the quality of the reconstructed data with the decoded SPIHT bits up to that point (i.e., the bitstream obtained by the proposed scheme is finer grained).

On the other hand, the increase in RD performance between the pair of plots ▲ and ● for the diffuse texture or the surface coefficients for a particular k was very small. This indicates that the bits decoded between points ▲ and ● did not include important information for the reconstruction. The reason for this behavior is apparent from the characteristics of the SPIHT algorithm.

To further understand this tendency, we conducted another experiment. The results are shown in **Fig. 5**. These results are only for the R signal of the horse 1 data set. The sequence of plots connected by lines ■, ▲, ●, and × all represent the performance of our method, and each sequence of plots represents results decoded from the same bitstream. The difference between the encoding methods for each plot sequence is the point of truncation of the SPIHT bitstream at the encoder. For plots ■, SPIHT encoding was carried out losslessly (i.e., with no truncation). For plots ▲, SPIHT encoding was terminated by not encoding the least significant bits (LSBs). For plots ●, the LSB and the second LSB were not encoded. For plots ×, the LSB, second LSB, and third LSB were not encoded.

For the plot sequences ■, ▲, and ●, the more the SPIHT bitstream was truncated, the better the RD performance was. This is because the truncated data did not contain important information for reconstruction, and significant numbers of negligible bits were eliminated from the bitstream. On the other hand, plot sequence × indicates a significant degradation in RD performance. This indicates that the SPIHT bits were over-truncated. Furthermore, adding up to eight pairs of the view basis vector and the surface coefficient set did not recover the RD performance, compared with the other plot sequences. For these plot sequences, only five pairs were used. In these experiments, we applied the same rule of SPIHT encoding termination for every pair of the view basis vector and the surface coefficient set. Since the probability for high-magnitude wavelet transform coefficients tends to be lower for the later pairs in the bitstream, we suspect that changing the termination rule, depending on the pair can increase the RD performance. We intend to investigate this bit allocation problem in future.

Finally, we will present examples of the rendered images that correspond to the results discussed above. The rendering algorithm presented in [6] has not yet been implemented. Instead, our current implementation uses the following algorithm. Every time the view of the virtual camera is changed, at every surface sampling point u_i ($i = 0, 1, .., I - 1$) a view-dependent blending of the color presented in [14] is done to obtain one color per sampling point u_i. Then the mesh defined by the subdivision connectivity of the surface sampling points u_i (i.e., the mesh that treats u_i as vertices) is rendered, and the colors evaluated for each surface sampling point are interpolated for the whole surface by the procedure used in

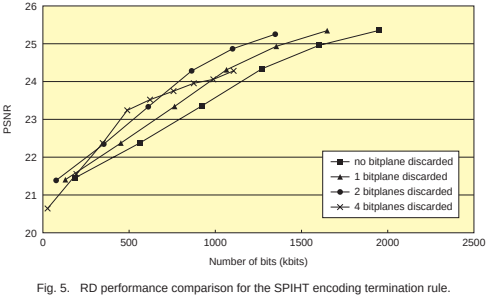

Fig. 5. RD performance comparison for the SPIHT encoding termination rule.

Gouraud shading. **Figure 6** is a rendered image of the uncompressed SLF. **Figure 7** is a rendered image of the result corresponding to the first plot on the left as ■ in Fig. 3. **Figure 8** is a rendered image of the result corresponding to the first plot on the right as ■ in Fig. 3. We confirmed that the result in Fig. 7 showed a diffuse component-like texture being recovered. In Fig. 8, adding the pair of view basis vector and surface coefficient set to the reconstruction added a specular component-like color to the rendered image.

5. Conclusion

In this paper, we described our progressive SLF compression scheme. This scheme uses a compressed representation of the SLF obtained using the view basis and the surface coefficients to enable on-the-fly reconstruction. The pairs of view basis vector and surface coefficients set are placed in the bitstream in the order of importance, and the surface coefficients are progressively encoded to obtain a finer-grained bitstream. Progressiveness of the bitstream allows the user to decode the SLF without waiting for the whole bitstream to be transmitted, in a low bandwidth environment. Also, rendering can be done without consuming much memory, allowing the user to render with a high-resolution SLF, or with multiple SLFs simultaneously.

Our future plans include investigating 1) a bit allocation strategy for the pair of view basis vector and surface coefficient set and 2) a rendering algorithm that further exploits conventional rendering-accelerating graphics hardware compared with the algorithm proposed in [6].

6. Acknowledgments

The horse 1 dataset was provided by Intel Corporation [15].

References

[1] "Application and Requirements for 3DAV," document N5877 MPEG Trondheim Meeting, 2003.
[2] S. B. Kang, "A Survey of Image-Based Rendering Techniques," VideoMetrics, Vol. SPIE2641, pp. 2-16, 1999.
[3] G. Miller, S. Rabin, and D. Ponceleon, "Lazy decompression of surface light fields for precomputed global illumination," Eurographics Rendering Workshop, pp. 281-292, 1998.
[4] M. Levoy and P. Hanrahan, "Light field rendering," Proc. SIGGRAPH'96, Los Angeles, U.S.A., pp. 31-42, Aug. 1996.
[5] M. Magnor, P. Pamanathan, and B. Girod, "Multi-View Coding for Image-Based Rendering Using 3-D Scene Geometry," IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, No. 11, pp. 1092-1106, 2003.

Fig. 6. Rendered image for the uncompressed SLF.

Fig. 7. Rendered image for the reconstructed SLF where the result corresponds to the first plot on the left in Fig. 3.

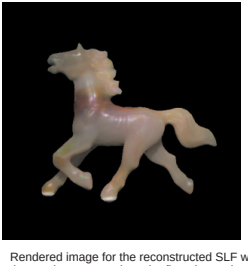

Fig. 8. Rendered image for the reconstructed SLF where the result corresponds to the first plot on the right in Fig. 3.

[6] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," Proc. SIGGRAPH'00, New Orleans, Louisiana, U.S.A., pp. 287-296, July 2000.
[7] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation," IEEE Trans. on Pattern Analysis and Machine Intelligence , Vol. 11, No. 7, pp. 674-693, 1989.
[8] P. Schroder and W. Sweldens, "Spherical Wavelets: Efficiently Representing Functions on the Sphere," Proc. SIGGRAPH'95, Los Angeles, U.S.A., pp. 161-172, Aug. 1995.
[9] W. Lee, W. Sweldens, P. Schroder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution adaptive parameterization of surfaces," Proc. SIGGRAPH'98, Orlando, U.S.A., pp. 95-104, July 1998.
[10] A. Said and W. Pearlman, "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," IEEE Trans. Circuits and Systems for Video Technology , Vol. 6, No. 3, pp. 243-250, 1996.
[11] A. Khodakovsky, P. Schroder, and W. Sweldens, "Progressive Geometry Compression," Proc. SIGGRAPH'00, New Orleans, Louisiana, U.S.A., pp. 271-278, July 2000.
[12] B. Grung and R. Manne, "Missing values in principal component analysis," Chemometrics and Intelligent Laboratory Systems, Vol. 42, pp. 125-139, 1998.
[13] D. Zorin, P. Schroder, and W. Sweldens, "Interpolating Subdivision for Meshes with Arbitrary Topology," Proc. SIGGRAPH'96, Los Angeles, U.S.A., pp. 189-192, Aug. 1996.
[14] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," Proc. SIGGRAPH'01, Los Angeles, U.S.A., pp. 425-432, Aug. 2001.
[15] http://www.intel.com

Masaki Kitahara
Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.
He received the B.E and M.E degrees in industrial and management systems engineering from Waseda University, Tokyo in 1999 and 2001, respectively. He joined NTT in 2001 and has been engaged in R&D of model-based data compression for image based rendering and H.264 video compression algorithms. His research interests include signal processing methods for 3D applications, data compression, and hardware-assisted rendering algorithms for image-based rendering, and mesh parameterization. He is a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE).

Hideaki Kimata
Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.
He received the B.E. and M.E. degrees in applied physics from Nagoya University, Nagoya, Aichi in 1993 and 1995, respectively. In 1995, he joined NTT Human Interface Laboratories, where he has been engaged in R&D of low bitrate video coding and error resilient video coding algorithms and visual communication systems. His research interests also include free viewpoint video coding and pre- and post-processing for video coding. He acts as a co-chair of MPEG 3DAV AHG. He is a member of IEICE and the Institute of Image Information and Television Engineers of Japan(ITE).

Kazuto Kamikura
Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.
He received the B.E. and M.E. degrees in electrical engineering from Tokyo Science University, Tokyo in 1984 and 1986, respectively. He received the Ph.D. degree in electrical engineering from the University of Tokyo, Tokyo in 2000. He joined NTT in 1986 and has been engaged in R&D of video coding systems. His current research interests include digital image processing and video sequence coding. He is a member of IEICE and ITE.

Yoshiyuki Yashima
Senior Research Engineer, Visual Media Communications Project, NTT Cyber Space Laboratories.
He received the B.E., M.E., and Ph.D degrees in electronics engineering from Nagoya University, Nagoya, Aichi in 1981, 1983, and 1998, respectively. In 1983 he joined the Electrical Communications Laboratories, Nippon Telegraph and Telephone Public Corporation (now NTT), Kanagawa, where he has been engaged in R&D of high-quality HDTV signal compression, MPEG video coding algorithms, and lossless image coding systems. His research interests also include pre- and post-processing for video coding, processing of compressed video, compressed video quality metrics, and image analysis for video communication systems. He is a member of the IEEE Signal Processing Society, the Information Processing Society of Japan, IEICE, and ITE.