SKIN-OFF: REPRESENTATION AND COMPRESSION SCHEME FOR 3D VIDEO

Hitoshi Habe

Graduate School of Engineering,
Kyoto University
Kyoto 606-8501, Japan
habe@media.kyoto-u.ac.jp

Yosuke Katsura, Takashi Matsuyama

Graduate School of Informatics,
Kyoto University
Kyoto 606-8501, Japan
{katsura,tm}@vision.kuee.kyoto-u.ac.jp

ABSTRACT

3D video records dynamic 3D visual events as is. The application areas of 3D video include wide varieties of human activities. To promote these applications in our everyday life, a standardized compression scheme for 3D video is required. In this paper, we propose a practical and effective scheme for representing and compressing 3D video named *skin-off*, in which both the geometric and visual information are efficiently represented by cutting a 3D mesh and mapping it onto a 2D array. Our skin-off scheme shares much with geometry videos, proposed by Hoppe et al. However, while geometry videos employ the 3D surface shape information alone to generate 2D images, the skin-off scheme we are proposing employs both 3D shape and texture information to generate them. This enables us to achieve higher image quality with limited bandwidth. Experimental results demonstrate the effectiveness of the skin-off scheme.

1. INTRODUCTION

By embedding a group of video cameras in the real world, we can digitize and record dynamic 3D visual events as is: human dances & sports, animal behaviors, and natural environments[8]. We call this type of image media 3D video. The application areas of 3D video include wide varieties of human activities: entertainment (e.g. 3D game and 3D TV), education (e.g. 3D animal picture books), sports (e.g. sport performance analysis), medicine (e.g. 3D surgery monitoring), culture (e.g. 3D archive of traditional dances) and so on. To promote applications of 3D video in our everyday life, a standardized compression scheme for 3D video is required. In this paper, we propose a practical and effective scheme for representing and compressing 3D video, in which visual and 3D geometric information are represented by 2D arrays.

3D video data consist of a time-varying 3D surface and video data mapped onto the surface. The key idea of our scheme rests in that 1) by cutting open the closed 3D surface, we can map (i.e. skin-off) the 3D surface onto a 2D plane, 2) project multi-view video data associated with the 3D surface onto the 2D plane, and 3) apply conventional video compression to the projected 2D video. Fig.1 depicts this scheme. This scheme is very practical since we can employ existing 2D video compression devices and software.

This work is supported by Grant-in-Aid for Scientific Research of MEXT under the contraction of 13224051 and the Research Project for Development of High Fidelity Digitization Software for Large-Scale and Intangible Cultural Assets of MEXT. Omni-directional video data is provided by 3DMM Committee, JEITA and Prof. Yokoya's Laboratory, NAIST, Japan

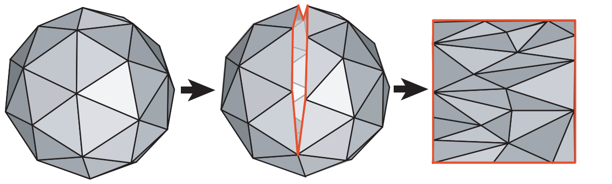


Fig. 1. "Skin-off" operation for representing 3D video data by a 2D array

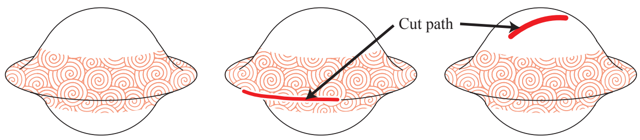


Fig. 2. Cut path finding using texture information

Hoppe et al. proposed geometry images for representing the geometrical structure of a 3D mesh[3] and geometry video for a sequence of 3D meshes[2]. Our skin-off scheme shares much with geometry video with regards to cutting 3D meshes and mapping it onto a 2D domain. However, while geometry images and video employ the 3D surface shape information alone to generate 2D images, the skin-off method we are proposing employs both 3D shape and texture information to generate them. Fig.2 illustrates the intuitive comparison between two schemes, where a given 3D mesh has high-curvature areas, on which texture patterns are mapped. While, to generate the geometry image, a cut path for unfolding the 3D mesh is selected so that it passes through the high-curvature area, our skin-off scheme selects the cut path which passes areas where no prominent texture exists. Since meshes adjacent to the cut path are stretched or shrunk extremely, more texture distortion occurs when we use the geometry image scheme. In general, when we visualize 3D video using the decompressed data, noticeable artifacts will appear as the distortion of the texture mapped on a surface. That is, 3D shape distortions may not be perceived clearly if no prominent texture exists on distorted surfaces, while textural distortions can be well perceived even if 3D shape is accurately preserved. Our skin-off scheme employs such texture information to transform 3D video onto 2D images and enables us to achieve higher image quality with limited bandwidth.

In the first part of this paper, we investigate the skin off scheme for omni-directional video as a special case of 3D video, in which the 3D surface is fixed and can be assumed to be a sphere. Then the latter part of the paper addresses general case of 3D video, where the shape of the 3D surface changes dynamically depending on 3D object actions.

2. SKIN-OFF: A SCHEME FOR 3D VIDEO COMPRESSION

2.1. Camera Configuration and 3D Video Data

Spatial camera configurations for recording dynamic visual events can be categorized into three types (Fig.3): parallel view (i.e. cameras are aligned along a straight line to observe a wide scene), convergent view (i.e. all cameras are looking at an object from different viewpoints) and divergent view (i.e. cameras are looking around the scene to generate its omni-directional view).

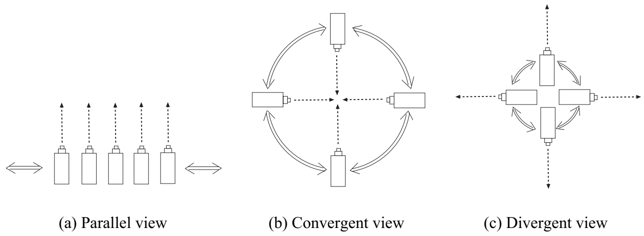


Fig. 3. Spatial camera configurations for recording dynamic visual events

In the convergent and divergent views, a group of captured video data can be mapped onto a common 3D surface: for example, a 3D human body surface for the former and a spherical surface for the latter. We call such image media 3D video. That is, 3D video is generated from a group of geometrically calibrated and temporally synchronized multiple video streams and can be represented by 1) a 3D (dynamically deforming) surface representing 3D shape of a recorded object or an observed scene, onto which video data are mapped and 2) a group of multi-view video data representing dynamically changing appearances of a recorded object or an observed scene.

In the convergent view, both the shape of the surface and the visual data on the surface change dynamically. Our skin-off algorithm for 3D video handles such dynamically changing surface and video data. As for the divergent view, i.e. omni-directional video, the shape of the surface is fixed and can be assumed to be a sphere and the visual data mapped on the sphere is dynamically changing. Since this corresponds to a special case of 3D video, we can apply a specialized version of the skin-off algorithm designed for omni-directional video.

In this paper, we address these two types of skin-off algorithms which share our key idea, i.e. cutting open the 3D meshes and employing 2D arrays for representing the geometric and visual information of 3D video,

2.2. Skin-off Operations

As described in Sec.1, the key idea of our scheme rests in that 1) by cutting open the closed 3D surface, we can map (i.e. skin-off) the 3D surface onto a 2D rectangular plane, 2) project video data associated with the 3D surface onto the 2D plane, and 3) apply conventional video compression to the projected 2D video representing the geometry and video data. This scheme is very practical since we can employ existing 2D video compression devices and software.

What we have to study technically is the problem of efficiency. That is, what types of skin-off operations are effective for 3D video data? More specifically, a critical problem in this skin-off operation rests in how to select appropriate path(s) for cutting off the 3D surface and a suitable function to map the 3D surface onto a 2D plane; the cut operation introduces distortions along its path(s). The cut path and mapping function problem can be classified into: 1) intra-frame optimization problem: how to find the cut path and the mapping function with the least distortions and 2) inter-frame optimization problem: how to coordinate multiple cut paths and mapping functions for temporal sequence of 3D video frames. In this paper, we confine ourselves to the first problem.

2.3. Related Works and Our Contributions

Since 3D video shares much with 3D animation, we should compare our method with those devised for 3D animation and clarify our originality.

Taubin and Rossignac [14] introduced a method to compress the connectivity of a static mesh. It employs predictors to encode the vertex positions effectively. Although this method works very well for encoding connectivity information, geometry information is not as highly compressed and a codec has to be newly designed for this method. Other researchers proposed similar approaches which compress the connectivity of a static mesh[6, 7, 1, 4, 15]. These methods employ temporal or spatial redundancy and accomplish high compression efficiency for connectivity. However a codec has to be newly designed for these methods too. This would be a disadvantage with respects to the practical usability.

Hoppe et al. proposed geometry images for representing the geometrical structure of a 3D mesh[3, 9] and geometry video for a sequence of 3D meshes[2]. The connectivity between vertices is recorded on the geometry image implicitly, which enables us to transmit all 3D video data, i.e. the geometry and video data, with existing codecs. As described in Sec.1, our skin-off scheme shares much with the geometry video: cutting 3D meshes and mapping it onto a 2D domain. However, while the geometry images and video employ the 3D surface shape information alone to generate 2D images, the skin-off method we are proposing employs both 3D shape and texture information to generate them. This allows us to obtain the decompressed image with higher quality.

As for the texture information, Hoppe et al. introduced the signal specialized parameterization[10, 11], in which, in order to reduce memory requirements for texture mapping a model, the signal-stretch metric was derived. The metric integrates signal, e.g. texture or surface normal, approximation error over the surface. By using this metric, more texture samples are allocated to mesh regions with greater texture variation and the quality of rendered image is improved.

The contributions of this paper can be summarized as 1) introducing the concept of skin-off as the common scheme for compressing 3D video, and 2) clarifying the validity and effectiveness of the skin-off operation, where the texture information plays a major role in its processing.

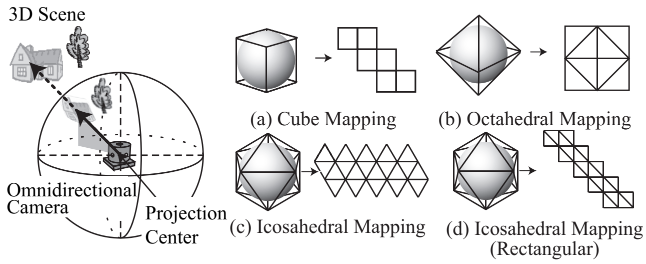


Fig. 4. Polyhedral mappings for representing omni-directional video data on a 2D plane

3. SKIN-OFF FOR OMNI-DIRECTIONAL VIDEO

As depicted in Fig.4, omni-directional video data can be mapped on a spherical screen, whose shape stays fixed. This type of image media can be regarded as a special case of 3D video. In this case, we need not transmit any geometrical information, i.e. the shape of the sphere, in every frame. Therefore we introduce a specialized version of the skin-off operation for omni-directional video

The specialized skin-off operation consists of 1) employ a polyhedral screen, 2) project video data on the spherical screen onto the polyhedral screen, 3) unfold the polyhedral screen onto a 2D plane by cutting along certain path, and 4) apply existing video compression method for the 2D video[5]. In this algorithm, the geometry of polyhedral screen and the cut path are determined in advance and not transmitted. Though the spherical wavelet[12] can be used for representing omni-directional video, the specialized skin-off operation is more practical since the existing video codec can be applied.

Then, the question to be answered is which type of polyhedron should to be used for coding omni-directional video. Among a wide variety of polyhedra, regular polyhedra should be used to model omni-directional video; non-regular ones may introduce non-isotropic distortions depending on viewing directions. Then, since the polyhedron which has large number of surfaces approximates a sphere well, resolution distribution on such a polyhedral surface will be uniform and higher compression efficiency can be achieved.

In Sec.5, we conduct experiments to compare the performance of various kinds of polyhedral screens and unfolding operation for compressing omni-directional video.

4. SKIN-OFF FOR 3D VIDEO

This section describes the skin-off scheme for generic 3D video, which has dynamically changing 3D mesh and video data mapped onto it. As noted above, the key idea of our skin-off scheme rests in the utilization of the texture distribution on the surface. In this section, we will focus on how to incorporate such texture information into the skin-off operation.

First of all, we introduce a metric for representing the "density" of texture on the surface. The metric can be defined as:

$$T(s) = \frac{1}{A_s}\int_s \sqrt{d_x(p)^2 + d_y(p)^2}\mathrm{d}p, \qquad (1)$$

where s denotes the surface patch of the 3D object, A_s the area of s, p a point on s, and $d_x(p),\ d_y(p)$ the spatial differentials at p.

4.1. Cut-path Finding

First, we study the cut-path finding problem: how to select appropriate cut-path(s) for cutting open a 3D mesh. When "extrema" of a 3D mesh is mapped onto a 2D plane without a cut path(s) nearby, the geometrical distortion becomes very large. To avoid this distortion and obtain efficient geometry images, Hoppe et al. employ an iterative algorithm which searches extrema of mesh and augments the current cut path so that it passes through the extrema[3].

Besides this geometrical optimization, our skin-off scheme exploits the texture distribution on the surface. When we augment the current path, we try to find edges whose adjacent surfaces have lower texture density $T(s)$ and select a new cut path so that it passes through them. The rationale behind this is that meshes adjacent to the cut path, which correspond to the border of the geometry image, are stretched or shrunk extremely and the texture distortion will become large on such region. We define the texture density associated with the edge as $D(e) = T(s_1) + T(s_2)$ where $s_1,\ s_2$ denote the surfaces which share the edge e.

4.2. Mapping a 3D mesh onto a 2D plane

After the cut path is selected, we have to investigate how to determine the mapping function from a 3D mesh onto a 2D plane, i.e. parameterization. In this phase, our goal is that the texture and geometry information of 3D video is allocated efficiently in the limited 2D plane.

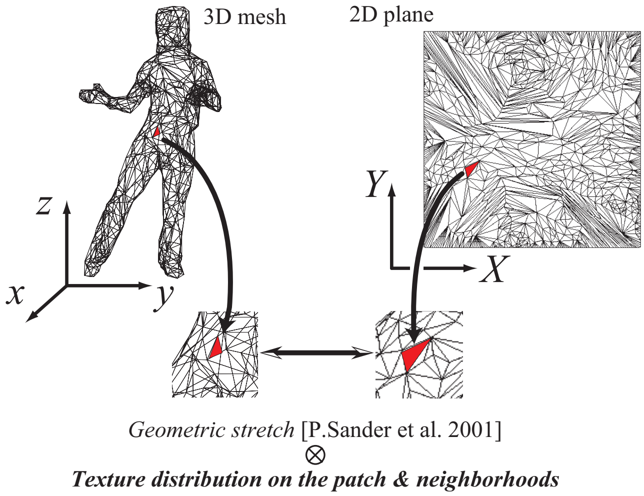


Fig. 5. Optimization of mapping function from a 3D mesh to a 2D plane

To obtain the geometry image, the geometric stretch metric[10] $m_{\mathrm{G}}(s)$ is introduced for measuring how the triangle mesh is stretched after being mapped on a 2D plane. In addition to the geometric stretch, we employ the *texture-based metric* described below for incorporating the texture information into the optimization of the mapping function (Fig.5). In the optimization process, each vertex is moved to minimize the sum of the integrated metric m at all surfaces on the 3D mesh. The metric m is designed to 1) minimize the weighted sum of $m_{\mathrm{G}}(s)$, where the weight is determined by the texture density on the surface, and 2) avoid discontinuity of the stretch. To define $m(s)$, we introduce 1) the texture density metric $m_{\mathrm{T}}(s)$ and 2) the stretch direction metric $m_{\mathrm{S}}(e)$.

The texture density $m_{\mathrm{T}}(s)$ is the weight factor of the geometric stretch $m_{\mathrm{G}}(s)$ and quantify the texture density at the surface s. Through experiment, we have found that only taking account of the texture density $T(s)$ on the surface s results in discontinuity of the stretch. This causes jitter on the rendered image. Hence, $m_{\mathrm{T}}(s)$ is defined as:

$$f_{\mathrm{T}}(s) = \sum_{t \in N(s)} \alpha(t,\, s) T(t), \tag{2}$$

where $N(s)$ denote a set of surface patches neighboring the surface patch s, α takes higher value when t is close to s.

Experimental results indicate that when the direction of the stretch drastically changes between the adjacent surface patches, noticeable artifacts arise on the border of such patches. To reduce such artifacts, we introduce the metric $m_{\mathrm{S}}(s)$ which measures the continuity of the stretch direction.

The intuition for this metric is described in Fig.6, where adjacent two 3D patches are mapped onto a 2D plane in different manners. In the case of (a), the relative angle θ between the two stretch directions, which are indicated with dotted arrows, are almost equal to that of the 3D mesh. In this case, the sampling rate on the surface does not change so much and the distortion on the shared edge will be reduced. On the other hand, in the case of (b), the angle θ is changed from that in a 3D domain. Discontinuity arises on the border of these two surfaces and causes the distortion on the rendered image.

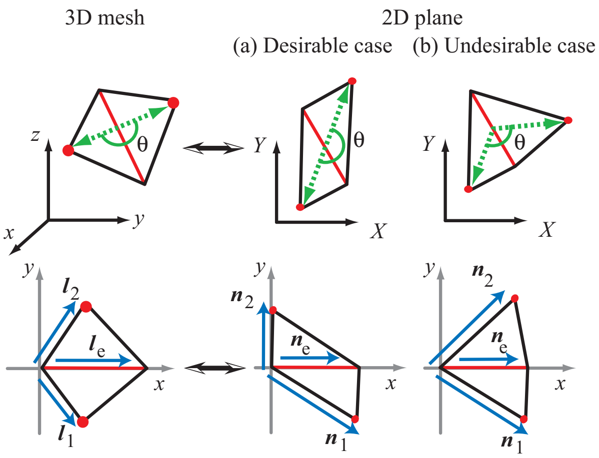


Fig. 6. Stretch direction continuity between adjacent patches

Based on this rationale, $m_{\mathrm{S}}(s)$ is derived as: 1) project two 3D patches onto a 2D plane, 2) rotate two patches so that the shared edge corresponds to x-axis as illustrated in the lower row of Fig.6, 3) calculate $m_{\mathrm{S}}(e)$:

$$m_{\mathrm{S}}(e) = \left|\left| \frac{1}{||\boldsymbol{l}_{\mathrm{e}}||}(\boldsymbol{l}_1 + \boldsymbol{l}_2) - \frac{1}{||\boldsymbol{n}_{\mathrm{e}}||}(\boldsymbol{n}_1 + \boldsymbol{n}_2) \right|\right|. \tag{3}$$

Minimizing this metric corresponds to keeping $\boldsymbol{l}_1 + \boldsymbol{l}_2$ and $\boldsymbol{n}_1 + \boldsymbol{n}_2$ equal to each other in a normalized domain.

At the last stage, we integrate these metrics, $m_{\mathrm{T}}(s)$, $m_{\mathrm{S}}(e)$ and the geometric stretch $m_{\mathrm{G}}(s)$, using the equation:

$$m = A \sum_{s} m_{\mathrm{T}}(s) \cdot m_{\mathrm{G}}(s) + B \sum_{e} m_{\mathrm{S}}(e), \tag{4}$$

(a) Data set 1 (15fps, About 4.7Mppf)
(b) Data set 2 (30fps, About 0.7Mppf)

Fig. 7. Test data sets of omni-directional video

where A and B are scale factors for balancing these two constraints.

4.3. Skin-off Algorithm

By incorporating our solution for the cut path and mapping function finding problem, we can summarize the skin-off algorithm as;

1. Select initial cut-path for transforming the surface of arbitrary genus into a topological disk. We employ the selection algorithm adopted in geometry images.
2. Based on the current cut path, optimize the mapping function. We apply the hierarchical optimization algorithm described in [11, 3], which simplifies a given mesh to form a progressive mesh representation, then, apply coarse-to-fine optimization approach to avoid local minima. We employ the metric m instead of the sum of geometric stretch and optimize the position of each vertex to minimize the metric m in each iteration.
3. Using the current mapping function, the cut-path is improved. Firstly, the extrema of a mesh is searched based on the geometric stretch from a 3D domain to a 2D domain. Then, we argument the current cut-path so that it passes through the extrema. Though the cut path extension algorithm is not clearly written by Hoppe et al, we accumulate the texture density $D(e)$ along several paths from the current path to the extrema. And we select a path which has least texture density $\sum D(e)$. Cutting along this path can be expected to produce the least distortion on the rendered image.
4. Step 2 and 3 are iterated until the mapping function no longer improves, i.e. until m converges.
5. Finally, we sample the 2D domain at grid points and obtain a 2D array on which the geometrical information, i.e. the vertex positions, and the texture information are stored. We can apply existing 2D video codec for it.

5. PERFORMANCE EVALUATION

In this section, we demonstrate the effectiveness of our skin-off scheme. After the evaluation of the specialized version for omni-directional video, we will show the effectiveness for 3D video in general.

5.1. Skin-off for Omni-directional Video

We employ two types of test data sets for evaluation of omni-directional video (Fig.7). The first one has 4.7 M pixel/frame, ppf in short, and the latter one has 0.7 Mppf. The framerate is 15 fps.

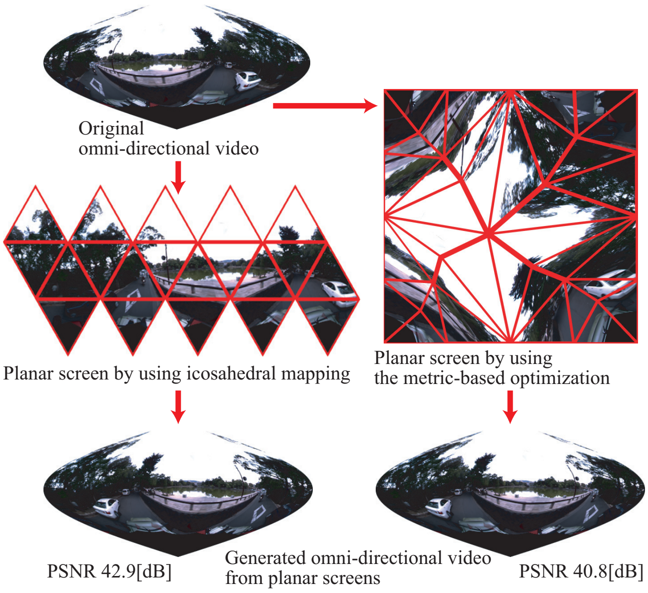


Fig. 8. Effectiveness of the specialized skin-off scheme

First, we evaluate the performance of the specialized skin-off algorithm for omni-directional video. As illustrated in Fig.8, omni-directional video data is mapped onto 2D arrays by using a icosahedral mapping and general skin-off method, i.e. the cut path finding and parameterization described in Sec.4. Then, from these 2D arrays, we generate omni-directional video data and compare the qualities of them. PSNR values are 42.9[dB] for the specialized skin-off, i.e. icosahedral mapping, and 40.8[dB] for the general skin-off. This result shows the effectiveness of the specialized skin-off for omni-directional video.

Then, we compare various kinds of polyhedral mappings described in Fig.4 and the cartographical mappings[13]. Note that Fig.4 (d) illustrates deformed icosahedral mapping which enables us to apply the block based coding methods as those for cube.

Evaluation method is as follows: 1) transform omni-directional video to 2D planar video using various kinds of mapping, 2) apply MPEG-4 compression/decompression and 3) generate an image viewed from a certain direction and compare it with its corresponding original one. Fig.9 shows the evaluation results. These results can be summarized as:

1. Compared with the cartographical mappings, the polyhedral mappings show higher image quality. This is because polyhedral mappings approximate a sphere well and a non-linear operation to map omni-directional data onto a cartographical screen causes the distortion.

2. Among polyhedral mappings, the regular icosahedral mapping shows higher image quality.

5.2. Skin-off for 3D Video

In this section, we evaluate the validity of our argument, that is, taking account of the texture information leads to higher image quality. Fig.10 illustrates artificially generated test data in which a non-uniform texture pattern is mapped onto a polyhedron with 80 faces.

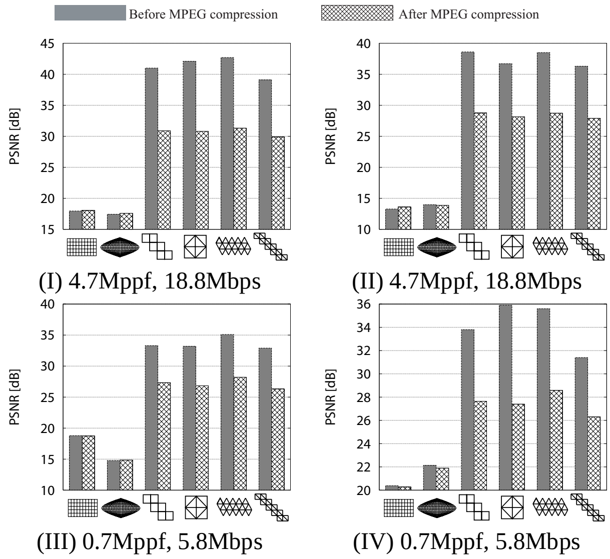


Fig. 9. Compression efficiency for omni-directional video

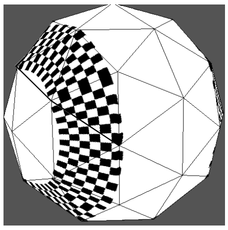


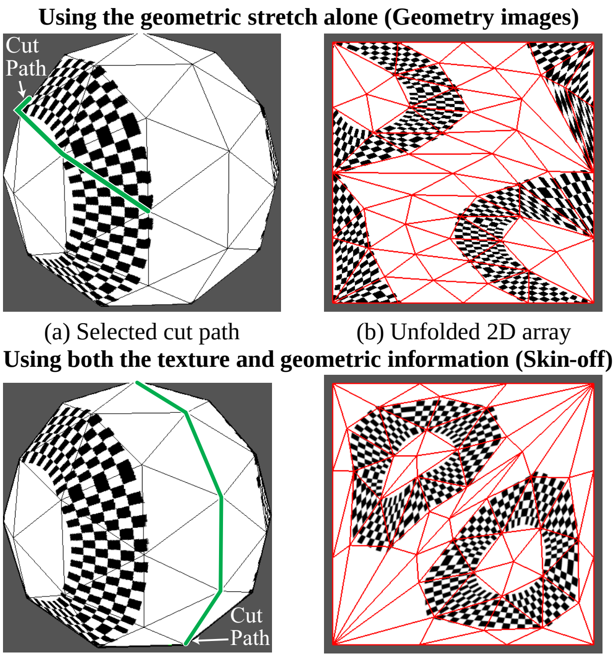
Fig. 10. 3D mesh and texture mapped onto it

We applied the geometry image scheme and obtained the cut path and the mapping function depicted in Fig.11 (a) and (b) respectively. In this experiment, since the 3D mesh has no "extrema", the cut path is randomly selected and can pass through areas where prominent texture exists (Fig.11 (a)). Then, by applying our skin-off scheme, the cut path illustrated in Fig.11(c)and the mapping function in (d) are obtained. Fig.11(c) shows that the skin-off scheme selects the cut path not passing textured areas.

Fig.12 (a) shows the rendered image after applying the geometry image scheme using the geometric information alone in the cut-path and mapping function finding(Fig.11 (a), (b)). The distortion is observed in the upper-right area. Fig.12 (b) depicts the rendered image by using both the geometric and texture information(Fig.11 (c), (d)), in which no noticeable distortion is observed. This is because the surface mapped checker pattern is projected onto a larger area on the 2D plane by taking the texture distribution into account. This result demonstrates the validity of our skin-off scheme with regards to the incorporation of the texture information.

6. DISCUSSION AND CONCLUDING REMARKS

In this paper, we propose the skin-off scheme as a framework for representing and compressing 3D video. 3D video data consists of the geometry of a 3D mesh and the video data mapped on it. The skin-off scheme represents both the geometric and visual informa-



(a) Selected cut path (b) Unfolded 2D array

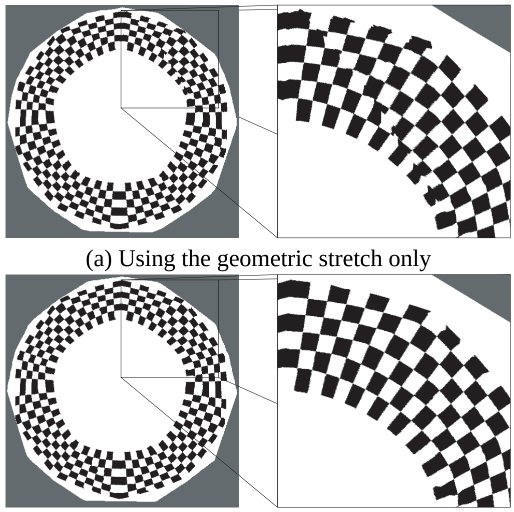
(c) Selected cut path (d) Unfolded 2D array

Fig. 11. Results of the cut path selection and the mapping onto a 2D array

tion efficiently by cutting the 3D mesh and mapping it onto a 2D plane. Since existing video codec can be applied for compressing these data mapped on a 2D plane, this scheme is very practical and can promote the application of 3D video in our everyday life.

Although a part of our implementation of the skin-off scheme is heuristic, experimental results demonstrate that it is promising. We will prove the effectiveness of our scheme based on the theoretical analysis. We will also conduct comprehensive evaluations using 3D video captured by multi-viewpoint cameras, for proving its effectiveness.

As mentioned above, though we put it beyond the scope of this paper, the inter-frame optimization problem, i.e. how to find an appropriate cut path and mapping function for a 3D video sequence, is also an important issue. Hoppe et al. discussed it in [2], but they have no definitive method to handle 3D video sequences. This would be because there is no invariant in the dynamically changing geometrical information. On the other hand, the texture information which we employ in the skin-off scheme can be invariant in time-varying 3D video sequences. This allows us to treat the texture information as the basis of compression. That is, the cut path and mapping function determined by using the texture information is expected to derive higher compression efficiency over all the sequence. This would be another advantage of the skin-off scheme. We will evaluate the effectiveness for the inter-frame optimization problem.



(a) Using the geometric stretch only

(b) Using both the texture and the geometric information

Fig. 12. Effectiveness of the texture information

7. REFERENCES

[1] P. Alliez and M. Desbrun. Progressive compression for lossless transmission of triangle meshes. *Proc. the 28th annual conference on Computer graphics and interactive techniques*, pp. 195–202, 2001.

[2] H. Briceno, P. Sander, L. McMillan, S. Gortler, and H. Hoppe. Geometry videos: A new representation for 3d animations. *ACM Symposium on Computer Animation 2003*, pp. 136–146, 2003.

[3] X. Gu, S. Gortler, and H. Hoppe. Geometry images. *SIGGRAPH 2002*, pp. 355–361, 2002.

[4] S. Gumhold and W. Straßer. Real time compression of triangle mesh connectivity. *Proc. the 25th annual conference on Computer graphics and interactive techniques*, pp. 133–140, 1998.

[5] H. Habe, K. Yamazawa, T. Nomura, and T. Matsuyama. Compression of omnidirectional video with polyhedral mapping. *Proc. the 18th Picture Coding Symposium of Japan 2003*, pp. 57–58, 11 2003 (in Japanese).

[6] L. Ibarria and J. Rossignac. Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity. *ACM Symposium on Computer Animation 2003*, pp. 126–135, 2003.

[7] J. E. Lengyel. Compression of time-dependent geometry. *Proc. the 1999 symposium on Interactive 3D graphics*, pp. 89–95, 1999.

[8] T. Matsuyama, X. Wu, T. Takai, and S. Nobuhara. Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video. *Computer Vision and Image Understanding*, 2004 (in Printing).

[9] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *SIGGRAPH 2003*, pp. 340–349, 2003.

[10] P. Sander, S. Gortler, J. Snyder, and H. Hoppe. Texture mapping progressive meshes. *SIGGRAPH 2001*, pp. 409–416, 2001.

[11] P. Sander, S. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. *Eurographics Workshop on Rendering 2002*, pp. 87–100, 2002.

[12] P. Schroder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. *SIGGRAPH 1995*, pp. 161–172, 1995.

[13] A. Smolic and D. McCutchen. 3dav exploration of video-based rendering technology in mpeg. *IEEE Trans. Circuits and Systems for Video Technology*, 14(3):348–356, 3 2004.

[14] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Trans. Graphics*, 17(2):84–115, 1998.

[15] C. Touma and C. Gotsman. Triangle mesh compression. *Graphics Interface*, pp. 26–34, 1998.