BIT ALLOCATION OF VERTICES AND COLORS FOR PATCH-BASED CODING IN TIME-VARYING MESHES

Toshihiko Yamasaki* † and Kiyoharu Aizawa‡ *,

*Dept. of Information and Communication Eng., The University of Tokyo ‡Interfaculty Initiative in Information Studies, The University of Tokyo, †MSR IJARC Fellow

ABSTRACT

This paper discusses bit-rate assignments for vertices, color, reference frames, and target frames in the patch-based compression method for time-varying meshes (TVMs). TVMs are nonisomorphic 3D mesh sequences of the real-world objects generated from multiview images. Experimental results demonstrate that the bit rate for vertices greatly affects the visual quality of the rendered 3D model, whereas the bit rate for color does not contribute to quality improvement. Therefore, as many bits as possible should be assigned to vertices, with 8–10 bits per vertex (bpv) per frame being sufficient for color. For interframe coding, the visual quality is improved in proportion to the bit rate of both vertices and color. However, it is demonstrated that the use of fewer bits (5~6 bpv) is sufficient to achieve a visual quality that matches the intraframe visual quality.

Index Terms—3DTV, time-varying mesh (TVM), interframe compression, vector quantization (VQ)

1. INTRODUCTION

For many years, three-dimensional television (3DTV) has been gaining much attention as a next-generation medium. Techniques for handling 3DTV can be divided approximately into two categories, namely image-based and model-based techniques.

For 3D scene generation, image-based rendering (IBR) [1][2] is popular because a scene via imaginary cameras can be obtained quickly without estimating the 3D shape of the objects. Moreover, a compression standard, called multiview coding, is already included in Annex H of the H.264/AVC standard.

Conversely, 3D geometric modeling [3]–[6] has some attractive features. First, there are many fewer cameras than for IBR. Second, 3D models can be seen from any viewpoint and provide a “freetr” viewpoint video than does IBR. Third, more flexibility in applications and processing is possible, such as the use of augmented reality.

TVM is a specific type of model-based 3D mesh sequence, being a sequence of nonisomorphic triangular meshes usually generated by the acquisition of a scene via multiple synchronized cameras. In most cases, for each mesh in the TVM, the vertex numbers, edge numbers, and connectivity information may vary between neighboring meshes [3]–[5]. Because of these characteristics, TVMs are very difficult to compress and only a few algorithms have been proposed so far [7]–[10].

In [10], we have proposed a patch-based matching algorithm for both interframe and intraframe coding of TVMs. In interframe compression, the 3D mesh in the target frame is divided into patches with similar surface area, which are compared with patches in the reference frame to find the most similar patch. The residual values, (Δx, Δy, Δz) for a vertex and (ΔR, ΔG, ΔB) for a color, are then encoded. In intraframe coding, the distributions of the raw values, (x, y, z) and (R, G, B), are encoded.

However, in [10], the assignment of bit rates to the vertices and the color that optimized the visual quality of the rendered 3D meshes was not discussed. In addition, the trade-off between bit rates for the reference frame and bit rates for the target frame was not clear. Therefore, in this paper, we present an intensive study that addresses these issues.

The remainder of this paper is organized as follows. Section 2 outlines the patch-based compression algorithm. Section 3 describes the details of the experiments, and concluding remarks are given in Section 4.
2. ALGORITHM FOR PATCH-BASED COMPRESSION

A simplified block diagram of the patch-based compression algorithm [10] is given in Fig. 1. The original 3D mesh model \( (M_i) \) is first divided into patches using segmentation via Dijkstra’s algorithm [11] so that their surface areas are almost the same. Then principal component analysis is applied to all the patches to place their centers of gravity at the origin of the world coordinate system and to adjust their orientation.

For the intraframe coding, the spatial redundancy of the data attached to each vertex is exploited. The VQ technique is employed for compressing the coordinates of the vertices and the color data. The codebooks are generated separately and they are sent to the decoder side. In addition, all residual data are entropy encoded. Although the spectral-compression-based method [12] offers superior performance for vertices, VQ is employed in this paper for simplicity.

For the interframe coding, the decoded reference frame \( (refM_i) \) and the original 3D mesh model in the target frame \( (M_j) \) are both divided into patches. Then patch-based matching is conducted between the patches in the target frame and the patches in the reference frame to find the nearest patches. Residual information is encoded using VQ. All the resultant data are entropy encoded.

3. EXPERIMENTS

The TVM data used in our experiments are summarized in Table 1. They were generated from multiple-view images taken by 22 synchronous cameras. The data were provided courtesy of Tomiyama et al. [3]. The sequence used in this paper is only one because the optimal bit allocation is the main concern. The validity of the compression algorithm itself using several sequences has already been demonstrated in another paper [10]. A 3D model for each frame is composed of three sets of data: coordinates of vertices, the connections to form triangular patches, and the color attached to each vertex. The color inside the patches is esti-
The PSNR for intraframe coding in the 3D space is shown in Fig. 2, and is calculated as follows.

For vertices, the PSNR is

$$PSNR_{\text{vertices}} = 10 \cdot \frac{(\text{diagonal of the bounding box})^2}{\sum_{i=1}^{N} (x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}$$

(1).

For color, the PSNR is

$$PSNR_{\text{color}} = 10 \cdot \frac{255^2}{\sum_{i=1}^{N} (R_i - R_r)^2 + (G_i - G_r)^2 + (B_i - B_r)^2}$$

(2).

Here, \((x_i, y_i, z_i)\) and \((R_i, G_i, B_i)\) represent the original values of the coordinates and the color of the \(i\)-th vertex, respectively, and \((x'_r, y'_r, z'_r)\) and \((R'_r, G'_r, B'_r)\) are their decoded values. \(N\) is the number of vertices in the model. The PSNR curve is obtained by averaging the PSNR values over all the frames. The codebook size is included in the bit rates. It is shown that the PSNR value increases in proportion to the bit rates for both vertices and color. The PSNR for the vertices is greater than that for color because the dynamic range is different.

In contrast, the PSNR of the rendered 2D images is very different from that in Fig. 2, as shown in Fig. 3. The graphs in Fig. 3 are drawn by fixing the bit rate either for vertices or for color and changing the other bit rate. Although the PSNR changes depend on the viewpoint, it is demonstrated that PSNR versus bit rate shows almost the same tendency from any viewpoint. In this paper, the viewpoint shown in Fig. 4 is employed. It is shown that the PSNR increases in accordance with the bit rate for the vertices, whereas the bit rate for color values does not contribute to the quality of the rendered images. The gradient of the PSNR is small in Fig. 3(b) and the PSNR value quickly saturates when the bit rate for color reaches 10 bpv (originally 24 bpv). This corresponds to an 8-bit codebook. This is because the color data are attached to the vertices and the quality of the rendered color texture strongly depends on the position of the decoded vertices. Therefore, accurate coding of the vertex position is more critical.

This phenomenon can also be observed in Fig. 4, which demonstrates the rendered images with different bit rates. The visual quality of the rendered images is rapidly improved as the bit rate for vertices increases. Therefore, we can conclude that more bits should be assigned to vertices than to color.

Fig. 5 shows the PSNR for the interframe encoded frames in the 3D space. The PSNR values are calculated in the same way as for the intraframe coding. Here, the bit rates for the vertices and for color in the target frame are set at the same value for simplicity. It is shown that the PSNR increases in proportion to the bit rate for the target frame. In contrast, the PSNR saturates when the bit rate for the reference frame reaches a certain value.

This phenomenon can also be observed in the rendered images, as shown in Fig. 6. The PSNR converges when the bit rate for the reference frame is around 10 bpv (the codebook size is 8 bits). For the target frame, however, the PSNR improves monotonically in accordance with the bit rate. If the bit rate for the reference frame is too low, the decoded reference frame is far from the original frame and the distribution of its vertices would become very different from the distribution for the target frame. Therefore, the bit rate in the target frame would be taken up by compensating for this error. When the bit rate for the reference frame reaches a certain value (in this case, about 10 bpv), the bit rates are used purely to encode the differences. It is also observed that a lower bit rate for the target frames than for the reference frames is sufficient to achieve the same PSNR. For instance, a quality of 30 dB is achieved at 14 bpv for
vertices but at 10 bpv for color in intraframe coding. Alternatively, the same quality can be achieved at 6 bpv for both vertices and color, provided the bit rate for the reference frames is above 10 bpv.

The rendered 3D models for the interframe coding are shown in Fig. 7. It is observed that the visual quality at lower bit rates is better than for intraframe coding.

Given that there are fewer frames for reference than for interframe coding and that the quality of the reference frame is proportional to the bit rate, the bit rates for reference frames should be made as large as possible.

4. CONCLUSIONS

This paper has presented an optimal bit-allocation strategy for vertex data, color data, reference frames, and target frames in encoding TVM sequences. It was demonstrated that coordinates of vertices were more important than color texture because the color data were also attached to vertices, and accurate decoding of the vertex data is more critical. It was also demonstrated that bit rates for the target frames that are half of those for the reference frames can achieve the same visual quality, showing the validity of the interframe coding technique used in the patch-based compression algorithm.

Part of this work was supported by the National Institute of Information and Communication Technology, Japan.

5. REFERENCES