PATTERN-BASED 3D MODEL COMPRESSION

CAI Kangying *JIANG Wenfei* *LUO Tao* *TIAN Jiang*

technicolor R&D

ABSTRACT

This paper presents an efficient method to 3D model compression based on repetition detection. The proposed *Pattern-Based 3D Mesh Codec* (*PB3DMC*) can achieve good rate-distortion performance on 3D models comprising multiple components. The repetition among constituent components is first exploited to generate a compact representation. An optimal bit allocation scheme is then proposed in order to compress the resultant "pattern-instances" representation. Experimental results show that *PB3DMC* yields a significant gain compared to the algorithms in MPEG's *Scalable Complexity 3D Mesh Coding* (*SC3DMC*) toolset, particularly for those models containing repetitive components. Furthermore, a benchmark for *PB3DMC* is built using 444 models. And *PB3DMC* is going to be published as an amendment of MPEG-4 standard.

***Index Terms*—** 3D model, Coding, repetitive component, bit allocation, MPEG

1. INTRODUCTION

The easy access to 3D modeling tools and rapid growth of online modeling communities make the quantity and complexity of 3D models increased continuously. And the widely used 3D modeling tools usually generate 3D models containing a significant number of components, especially when creating complex objects. Thus, efficient compression solutions are required to overcome the challenges in storage and transmission. For multi-component 3D models, it is straightforward to decompose them and compress each individual component using previous methods based on geometry primitives, such as vertex spatial position [1, 2], adjacency [3, 4, 5], local geometric features [6, 7].

However, there usually exists redundancy among the structures of constituent components. To improve the efficiency of 3D model coding techniques, the structural regularity can be exploited. Recently, symmetry detection on 3D models has received much attention [8, 9, 10]. Therefore, the detected symmetries can produce a compact representation of 3D model, such as a tree-structure [11, 8] or a hierarchy [12]. Digne et al. [13] proposed to compress point cloud by exploiting the self-similarity of the underlying shape. For image representation, the approaches to factoring repetitive content have also been developed. In [14], Wang et al. proposed to create a condensed epitome and a transform map such that all image blocks can be represented from transformed epitome patches. In our method, repetition is explored among the components of 3D model, which is a kind of geometric invariance under translation, rotation transformation or their combinations. The so-called "pattern-instances" representation derived from repetition detection is compact and beneficial to improve the performance of the proposed *PB3DMC*.

In order to encode the "pattern-instances" representation, an optimal bit allocation scheme is proposed in this paper. Different from the existing 3D model compression algorithms based on repetition detection [13, 15], *PB3DMC* determines the optimal quantization parameters automatically for various data in the repetition-aware representation. For instance, the quantization parameter of each rotation transformation is adaptively decided by the parameter of global quality and the scale of the corresponding component. To the best of our knowledge, it is the first time to concentrate on bit-rate allocation in the compression techniques based on repetition detection. And compared to the "instancing" techniques widely used by real-time engines for gaming, one important advantage of the proposed method is that the encoder requires no prior knowledge of the input 3D models.

In this paper, a benchmark is built for *PB3DMC* including 444 models provided by MyMultiMediaWorld.com [16]. As demonstrated in experimental results, *PB3DMC* makes a significant gain compared to the algorithms in MPEG's *SC3DMC* [17] toolset. And the bitstream format and decoder of *PB3DMC* are going to be published as one of the amendments in MPEG-4 Part 16 [18]. The 3D graphics group of MPEG (MPEG-3DGC) has standardized several compression techniques for 3D models, which are introduced in MPEG-4 Part 2, *Visual* (*3DMC*) and MPEG-4 Part 16, *Animation Framework eXtension* (*3DMC-Extension and SC3DMC*). *SC3DMC* is introduced in MPEG-4 Part 16 AMD 4 in year 2009, which consists of 5 different ways with different compression performances and complexities to enlarge the application domain of MPEG-4 3DG.

Thanks to the support from Institute of Software Chinese Academy of Sciences.

2. COMPRESSION ALGORITHM

The framework of PB3DMC encoder is as shown in Fig.1. The inputs are the 3D model to be encoded and one quantization parameter *QP* given by the user. Then the maximum reconstruction allowed, *MaxErr*, could be derived from QP and the boundingbox of the input model. The encoder consists of three steps, repetition discovery, repetition compression and instance verification.

The first step is achieved by pair-wise component comparison, which can be accelerated by grouping components using their feature descriptors. MaxErr is used as the threshold during the comparison. Then the input 3D model is transformed into the new representation consisting of *patterns*, *instances* and *unique parts*. Here a pattern refers to the representative geometry of a discovered repetitive structure, which has been aligned with the coordinate system. An instance is represented by the transformation matrix $\mathbf{M}$, from its corresponding pattern to this instance part, and the related pattern's ID. $\mathbf{M}$ is further decomposed into translation vector $\mathbf{T}(t_x, t_y, t_z)$ and rotation matrix $\mathbf{R}$, which is represented by the Euler angles (ψ, θ, ϕ) ($\theta \in \left[-\frac{1}{2}\pi, -\frac{1}{2}\pi\right]$, $\psi, \phi \in [-\pi, \pi]$). The unique parts are those not belonging to any repetitive structures.

In the second step, the patterns and unique parts can be encoded by any mature 3D model codec. $\mathbf{T}$ and (ψ, θ, ϕ) are quantized and further compressed by an entropy codec. QP_P, QP_T and QP_A, the quantization parameters of pattern, $\mathbf{T}$ and (ψ, θ, ϕ), are automatically determined by an optimal bit allocation scheme which could make the maximum reconstruction error meets the user requirement, i.e. MaxErr. Especially, to avoid unnecessary bit-rate loss, different instances use different values of QP_A, denoted as QP_{A_i}, which are related with the correspondent component scales. More details will be given later. For unique parts, QP is used as its quantization parameter. Furthermore, to guarantee the required reconstruction error, an instance verification step is followed. This step reconstructs every instance component to verify whether or not its maximum reconstruction error is less than MaxErr. Those fail to pass the test are compressed together with the unique parts discovered in the first step.

The compressed bitstream consists of QP, the boundingbox of the 3D model, the bounding-box of all translation vectors, the encoded unique parts, encoded patterns along with pattern IDS and encoded instance transformation matrixes along with instance IDs. Another advantage of the bit allocation scheme used by PB3DMC is that QP_P, QP_T and QP_{A_i} can be automatically decided by the decoder. Except the bounding-box of translation vectors, there is almost no bit-rate loss for recording different quantization parameters. The decoder decodes all the data and reconstructs instance components using the reconstructed patterns and transformation matrixes.

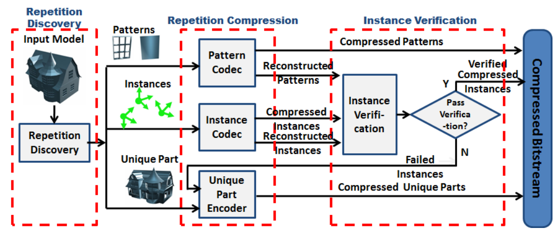

Fig. 1. PB3DMC encoder framework.

2.1. Bit Allocation Optimization

The optimal values of QP_P, QP_T and QP_{A_i} are automatically determined from QP by figuring out their contribution to the final reconstruction error. During the following discussion, the vertex-vertex error is used as the reconstruction error. Here the goal is to guarantee that the maximum reconstruction error is less than MaxErr.

Any vertex $\mathbf{v}$ on any instance component can be represented by

$$\mathbf{v} = \mathbf{R}\mathbf{p} + \mathbf{T}, \tag{1}$$

where $\mathbf{p}$ is the correspondent vertex on pattern , $\mathbf{R}$ and $\mathbf{T}$ are the rotation matrix and translation vector from $\mathbf{p}$ to $\mathbf{v}$.

Let $\mathbf{v}_d$ denote the reconstructed vertex. Then the reconstruction error of $\mathbf{v}$ can be calculated as:

$$\begin{aligned} \|\mathbf{v} - \mathbf{v}_d\| &= \|(\mathbf{R}\mathbf{p} - \mathbf{R}_d\mathbf{p}_d) + (\mathbf{T} - \mathbf{T}_d)\| \\ &\leq \|\mathbf{R}\mathbf{p} - \mathbf{R}_d\mathbf{p}\| + \|\mathbf{R}_d(\mathbf{p} - \mathbf{p}_d)\| + \|\mathbf{T} - \mathbf{T}_d\| \\ &\leq \|(\mathbf{R} - \mathbf{R}_d)\mathbf{p}\| + \|\Delta\mathbf{p}\| + \|\Delta\mathbf{T}\|, \end{aligned} \tag{2}$$

where $\mathbf{p}_d$, $\mathbf{R}_d$ and $\mathbf{T}_d$ are the reconstructed $\mathbf{p}$, $\mathbf{R}$ and $\mathbf{T}$. $\Delta\mathbf{p} = \mathbf{p} - \mathbf{p}_d$ and $\Delta\mathbf{T} = \mathbf{T} - \mathbf{T}_d$ are the reconstruction error of $\mathbf{p}$ and $\mathbf{T}$.

We have

$$\|(\mathbf{R} - \mathbf{R}_d)\mathbf{p}\| \leq c_\theta \Delta\theta_{max} \|\mathbf{p}_{max}\|, \tag{3}$$

where $\Delta\theta_{max}$ is the upper bound of quantization error of the three Euler angles and c_θ is an constant which can be estimated from experiments. In our experiments, $c_\theta = \sqrt{2}$.

From Eq. 2 and Eq. 3, the optimization goal here can be written as

$$c_\theta \Delta\theta_{max} \|\mathbf{p}\|_{max} + \|\Delta\mathbf{p}\|_{max} + \|\Delta\mathbf{T}\|_{max} = MaxErr, \tag{4}$$

where $\|\Delta\mathbf{p}\|_{max}$ and $\|\Delta\mathbf{T}\|_{max}$ are the upper bounds of the reconstruction error of pattern coordinates and translation vectors, and $\|\mathbf{p}\|_{max}$ is the scale of the pattern.

Intuitively, let the three items at the left side of Eq. 4 to be equal, i.e.

$$c_\theta \Delta\theta_{max} \|\mathbf{p}\|_{max} = \|\Delta\mathbf{p}\|_{max} = \|\Delta\mathbf{T}\|_{max} = \frac{MaxErr}{3}. \tag{5}$$

By using Eq. 5, the various quantization parameters can be decided from QP as follows.

QP_P is calculated by

$$QP_P = \lfloor 3c_{QP}QP \rfloor \tag{6}$$

where $c_{QP}\epsilon(0.0, 1.0)$ is a constant used to control the influence of the over-conservative estimation of the reconstruction to the final rate-distortion (*RD*) performance. In our experiments, $c_{QP} = 0.4$.

The patterns will be translated to the origin before compression. All patterns will be compressed together. Then

$$\|\Delta \mathbf{p}\|_{max} = 2^{-QP_P - 1} S_{P_{max}} \tag{7}$$

and

$$\|\Delta \mathbf{T}\|_{max} = 2^{-QP_T - 1} S_T \tag{8}$$

where $S_{P_{max}}$ is the scale of the pattern with the biggest bounding-box, and S_T is the the bounding-box of all translation vectors. Using Eq. 5, QP_T can be calculated by

$$QP_T = \lceil QP_P + \log_2 \frac{S_T}{S_{P_{max}}} \rceil. \tag{9}$$

For calculating QP_A, an important fact which can also be observed from Eq. 5 is that the distortion caused by quantizing Euler angles varies with the scale of the correspondent pattern. Thus, to achieve the same reconstruction error, relative small instance components need less accurate quantization of the rotation transformation than the big ones. To avoid unnecessary bit-rate loss, rather than calculating a single QP_A for all instances, QP_{A_i} is calculated for each instance using the correspondent pattern scale as follows.

As

$$\Delta\theta_{max} = 2\pi 2^{-QP_{A_i} - 1}, \tag{10}$$

using Eq. 5 and Eq. 7, QP_{A_i} is calculated by

$$QP_{A_i} = \lceil QP_P + \log_2 \frac{2\pi c_\theta S_{P_i}}{S_{P_{max}}} \rceil, \tag{11}$$

where S_{P_i} is the scale of the pattern corresponding to the i_{th} instance.

Furthermore, to make sure that the same values can be calculated by the decoder, QP_T and QP_{A_i} are given by

$$QP_T = \lceil QP_P + \log_2 \frac{S_T}{S_{P_{max_d}}} \rceil. \tag{12}$$

$$QP_{A_i} = \lceil QP_P + \log_2 \frac{2\pi c_\theta S_{P_{i_d}}}{S_{P_{max_d}}} \rceil, \tag{13}$$

where $S_{P_{max_d}}$ and $S_{P_{i_d}}$ are the scale of the reconstructed patterns. Then the decoder can automatically decided the quantization parameters after recovering all patterns.

3. EXPERIMENTAL RESULTS

Various experiments are performed to compare the *RD* performance of *PB3DMC* and *SC3DMC*. During all the experiments, the configuration of *SC3DMC* which generates the best performance is used. *PB3DMC* also uses *SC3DMC* with the same configuration to compress patterns and unique components. Multi-component 3D models with a wide range of complexity and topology types have been used in our experiments. *RMS* calculated by Metro [19] is used as the measure of the reconstruction error. The distortion is reported with the respect to the diagonal of the bounding box of the original 3D model.

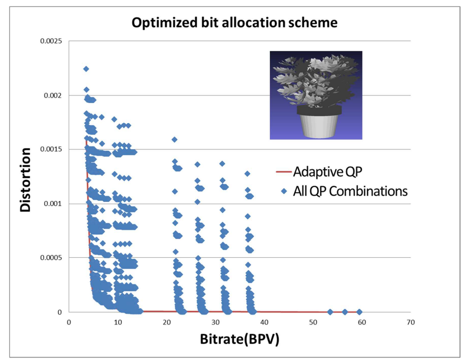

Fig. 2. Experiment to test the bit allocation scheme.

The first experiment is to investigate the optimal bit allocation scheme using model m1007 from Princeton Shape Benchmark [20]. Within the range of $[6, 22]$, *PB3DMC* using assigned values of QP_P, QP_T and QP_A is compared with *PB3DMC* using adaptive values of QP_P, QP_T and QP_A calculated from QP. As demonstrated in Fig. 2, the bit allocation scheme leads to a close-to-optimal performance among all the possible combinations of QP_P, QP_T and QP_A.

The *RD* gain of *PB3DMC* over *SC3DMC* is demonstrated in Fig. 3, using several 3D models from Princeton Shape Benchmark. It is clear that *PB3DMC* gains significant bit-rate saving when compressing 3D models with repetitive components. And the *RD* gain is more significant at low bit-rates because more repeated structures are discovered . As shown by m414's wireframe in Fig. 3(f), lots of multi-component 3D models, especially those CAD models, consist of lots of triangles in dramatically changing sizes and sharp features, which pose big challenges to traditional 3D model codecs. Moreover, the random position and orientation of the components of m973 and m1056 can hardly be exploited by traditional 3D model codecs. However, by using the new representation, these negative impacts on compression can be efficiently reduced.

Another advantage of *PB3DMC* is the decoding efficiency improvement. This is because that the decoding of transfor-

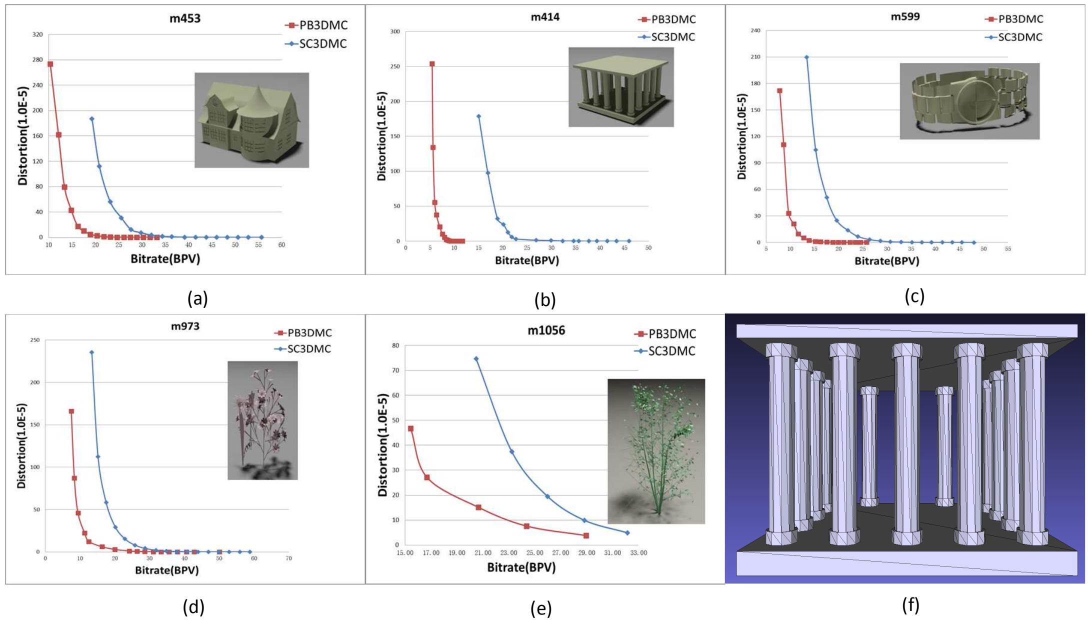

Fig. 3. The RD performance curves of 5 models from the Princeton Shape Benchmark. In the sense of the distortion is less than 0.001 , m453, m414, m599, m973 and m1056 have 70%, 95%, 82%, 73% and 75% vertices belonging to repetitive structures.

mation is usually more efficient than the decoding of vertices and triangles, which often requires complex computations. Experiments on 30 models from the Princeton Shape Benchmark is shown in Fig. 4, proving that *PB3DMC* achieves up to 60% decoding acceleration.

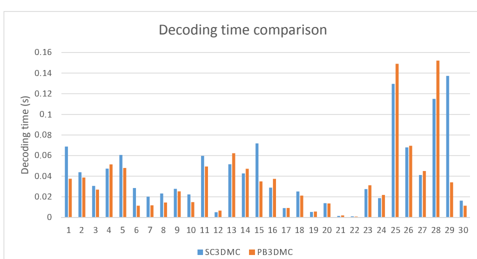

Fig. 4. Decoding efficiency comparison.

Finally, a benchmark is built for *PB3DMC* using 444 multi-component 3D models from MyMultiMediaWorld.com. The *RD* gain of *PB3DMC* over *SC3DMC* is as shown in Fig. 5. Except 7 models which *PB3DMC* performs a bit worse than *SC3DMC* ($< -10\%$), *PB3DMC* achieves significant gain, morn than 50% gain on more than 80% 3D models. Besides the efficient design of *PB3DMC*, this significant performance gain is also based on the fact that these 3D models are built for the common objects in the real world, such as plants and architectures, and contain a large number of components. Similar results are reported in [21].

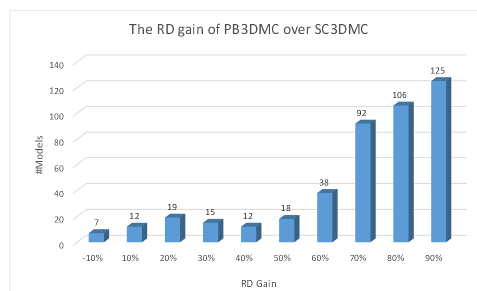

Fig. 5. The benchmark building result of *PB3DMC*.The numbers above the columns show the number of models achieving the corresponding RD gain.

4. CONCLUSIONS

In this paper, a *Pattern-Based 3D Mesh Codec* is presented to compress multi-component 3D models. The repetition among the components is exploited to reduce the redundancy, which benefit the performance of the proposed method. And an optimal bit allocation scheme is proposed to encode the compact representation derived from repetition detection. Compared with the previous standardized algorithms, a significant gain can be obtained using the proposed *PB3DMC*, which will be published as an amendment of MPEG-4 standard.

5. REFERENCES

[1] Jingliang Peng and C.-C. Jay Kuo, "Geometry-guided progressive lossless 3d mesh coding with octree (ot) decomposition," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 609–616, 2005.

[2] Pierre-Marie Gandoin and Olivier Devillers, "Progressive lossless compression of arbitrary simplicial complexes," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA, 2002, pp. 372–379, ACM Press.

[3] C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. of Graphics Interface*, 1998, pp. 26–34.

[4] C. L. Bajaj, V. Pascucci, and G. Zhuang, "Single resolution compression of arbitrary triangular meshes with properties," *Comput. Geom. Theory Appl.*, vol. 14, pp. 167–186, 1999.

[5] G. Taubin and J. Rossignac, "Geometric compression through topological surgery," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 84–115, 1998.

[6] S. Valette, R. Chaine, and R. Prost, "Progressive lossless mesh compression via incremental parametric refinement," *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1301–1310, 2009.

[7] Jingliang Peng, Yan Huang, C.Jay Kuo, Ilya Eckstein, and M. Gopi, "Feature oriented progressive lossless mesh coding," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2029–2038, 2010.

[8] Niloy J. Mitra, Leonidas J. Guibas, and Mark Pauly, "Partial and approximate symmetry detection for 3d geometry," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 560–568, 2006.

[9] Mark Pauly, Niloy J. Mitra, Johannes Wallner, Helmut Pottmann, and Leonidas J. Guibas, "Discovering structural regularity in 3d geometry," in *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, New York, NY, USA, 2008, pp. 1–11, ACM.

[10] Niloy J. Mitra, Mark Pauly, Michael Wand, and Duygu Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *EUROGRAPHICS State-of-the-art Report*, 2012.

[11] Patricio Simari, Evangelos Kalogerakis, and Karan Singh, "Folding meshes: hierarchical mesh segmentation based on planar symmetry," in *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, Aire-la-Ville, Switzerland, Switzerland, 2006, pp. 111–119, Eurographics Association.

[12] Y. Wang, K. Xu, J. Li, H. Zhang, A. Shamir, L. Liu, Z. Cheng, and Y. Xiong, "Symmetry hierarchy of man-made objects," *Comput. Graph. Forum*, vol. 30, no. 2, pp. 287–296, 2011.

[13] Julie Digne, Raphaelle Chaine, and Sebastien Valette, "Self-similarity for accurate compression of point sampled surfaces," *Computer Graphics Forum*, vol. 33, no. 2, Apr. 2014, Proceedings of Eurographics, to appear.

[14] H. Wang, Y. Wexler, E. Ofek, and H. Hoppe, "Factoring repeated content within and among images," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 14:1–14:10, 2008.

[15] Dinesh Shikhare, Sushil Bhakar, and Sudhir P. Mudur, "Compression of large 3d engineering models using automatic discovery of repeating geometric features," in *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*, 2001, pp. 233–240.

[16] B. Le Bonhomme, M. Preda, and F. Preteux, "Mymultimediaworld.com: A benchmark platform for 3d compression algorithms," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, Oct 2008, pp. 2700–2703.

[17] MPEG-3DGC, "w11455, iso/iec 14496-16 4th edition," in *93th MPEG meeting*, Geneva, Switzerland, 2010.

[18] Convenor of MPEG, "Mpeg press release, iso/iec jtc 1/sc 29/wg 11 n15328," in *MPEG112*, Warsaw, Poland, June 2015, MPEG.

[19] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, June 1998.

[20] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser, "The princeton shape benchmark," in *Shape Modeling International'04*, 2004, pp. 167–178.

[21] Oana Garoiu, Veronica Marin, Stefan Popa, Christian Tulvan, Ivica Arsov, Thomas Laquet, and Marius Preda, "m30438, preliminary results for pb3dmc benchmarking," in *MPEG105*, Vienna, Austria, July 2013.