Coding of Dynamic 3D Mesh Model for 3D Video Transmission

Jui-Chiu Chiang, Chun-Hung Chen, and Wen-Nung Lie

Department of Electrical Engineering
National Chung Cheng University, Chia-Yi, 621, Taiwan, ROC
{rachel,ieewnl}@ccu.edu.tw

Abstract. Recently, 3D video has gained increasing attention in multimedia field. The representation of 3D video is often based on dynamic 3D mesh model, which is reconstructed from multi-view video, plus surrounding texture information for rendering, so that arbitrary novel views can be synthesized accordingly. However, the dynamic 3D mesh model herein is not time-consistent, resulting in a difficulty in applying traditional mesh compression tools efficiently (e.g., MPEG-4 AFX 3DMC). In this paper, we modify the 3DMC algorithm for the coding and transmission of 3D video, taking its advantage of high coding efficiency for edge topologies and enhancing it with 3D motion estimation of vertices between two time-successive mesh models. Experiment results show that our method can reach about 30 times of compression ratio. Compared to MPEG-4 AFX 3DMC, under comparable reconstruction quality, our algorithm has a bit rate saving of about 20%~45%.

Keywords: 3D video, 3D mesh, 3D motion estimation.

1 Introduction

Nowadays, the development of multimedia video has already been promoted from 2D to 3D, or from single-view toward multi-views. With 3D video, observers are capable of seeing around an object by changing the viewing directions at their will. The observed views are no longer restricted to those captured by the really arranged cameras. For example, a dancer on the stage can be seamlessly looked around (or evenly zoomed-in/out) by an audience who controls a mouse to change the viewing direction [6]. This effect however relies on image projection of a 3D model or novel-view synthesis from multiply captured images. In a foreseeable future, 3D video will have more applications in education, art, entertainment, etc.

The representation of 3D video can be divided into two kinds: one is multi-view video plus depth information for each view [4], the other is dynamic 3D mesh model plus surrounding texture information for rendering [5]. Both kinds of methods need to arrange a number of inwards cameras for capturing the object views from several discrete directions. The former then estimates the disparity or depth information from any two adjacent views so that arbitrary novel views can be synthesized by using the DIBR (Depth Image-Based Rendering) technique, whereas the latter constructs a 3D

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 144–152, 2011.

model from all the captured views so that arbitrary views can be synthesized by projecting the 3D model onto an image plane, with textures being rendered thereon. Both methods take their own advantages and disadvantages. For example, multi-view video plus depth approach is advantageous of its system simplicity but is difficult in accurate disparity/depth estimation. On the other hand, the 3D mesh approach requires more cameras for accurate 3D model reconstruction, but benefits from flexibility in arbitrary view generation.

This work is a part of a 3D video system that adopts the approach of dynamic 3D mesh and texture rendering, focusing on the compression of the dynamic 3D mesh models that are reconstructed from a number of inwards cameras around the target objects. Though 3D mesh models have ever been widely used, the ones reconstructed in 3D video systems are different from those generated via tools of computer graphics. The most important is that topologies of the 3D mesh models (including the number of vertices, vertices comprising the triangular meshes, and number of triangles) most likely vary from time to time, presenting no correspondences between two 3D mesh models at successive time instants. This case, however, will not happen in computer-graphics-generated 3D mesh models.

The development of 3DMC (3D mesh coding) in MPEG-4 part 16 AFX (Animation Framework eXtension) has been mature for several years. This tool is however mainly developed for a single static model, but not for dynamic models generated/reconstructed at successive time instants. Though some techniques [7, 8] have been proposed to fill this gap, they are based on time-consistent dynamic 3D meshes (i.e., assuming that the vertex and edge sets of the 3D mesh models at successive time instants are kept preserved). As mentioned earlier, this assumption does not hold for 3D video applications. Theoretically, the MPEG-4 AFX tool can be applicable to the above-mentioned 3D video system by individually encoding the 3D mesh model reconstructed at each time instant. The encoding efficiency can be further improved by exploring the relation or redundancies between two time-successive mesh models.

It is the goal of this work to encode time-inconsistent dynamic 3D mesh models for 3D video transmission. The algorithm is essentially a modification of the MPEG-4 AFX, taking advantage of its high coding efficiency for edge topologies and enhancing it with accurate prediction of vertices between two time-successive mesh models.

2 Topological Surgery for 3D Mesh Model

There are many projects in MPEG-4 Part 16 AFX which concern about animation. Among them, 3DMC (3D mesh coding) was targeted for the compression of a single 3D mesh model. It adopts a “geometric compression through topological surgery (TS)” algorithm [1], which keeps the relation between meshes accurately and is capable of reaching a high compression performance.

The manner that 3DMC considers for a single mesh model can be named as "*intra-model*" coding. However, to handle dynamic 3D mesh models, the redundancy between each pair of time-successive mesh models should be taken into account for coding efficiency improvement. We name this "*inter-model*" coding, following the terminologies from the state-of-the-art video coding. The TS technique [1] is to dissect a mesh model into a spanned tree and then perform encoding of the resulting "vertex tree", "triangle tree", and "vertex coordinates". Among the above three quantities to be encoded, we first explore the inter-model redundancies for vertex coordinates in this work. For time-inconsistent dynamic 3D mesh models, the inter-model redundancies for the vertex trees and triangle trees remain still an open issue to the researchers. In one word, our algorithm follows the tree spanning and encoding procedures proposed in 3DMC, but modifies the encoding of vertex coordinates that constitute the most significant part of the resulting bit stream.

3 Inter-model Coding Based on 3D Motion Estimation

We borrow the concept of inter-frame 2D motion estimation from video coding for this inter-model vertex prediction, that is, "*3D motion estimation*" which predicts vertex coordinates of the current (time) model from the previous (time) model and encodes the residuals. It is inefficient for each vertex to have a prediction parameter (e.g., 3D transform parameters) for calculating the residuals. Rather, we group vertices to adopt a limited set of 3D transform parameters. It is observed that transformation between successive (-time) 3D mesh models is often non-rigid (e.g., motions of the human's body and limbs might not be consistent). Hence, k-means clustering algorithm (in this work, k=5, considering human's body and 4 limbs) is adopted to partition the vertices of each model into k groups. Then the well-known ICP (Iterative closest point) algorithm [2] is used to align each group of vertices with those (the whole set) in the previous reconstructed (after decoding) mesh model $\widetilde{F}_{n-1}$. The estimated 3D transform parameters by using the ICP algorithm are regarded as the 3D motion parameter by which vertices of each cluster can be nearly aligned (or, closest to) with one of those in $\widetilde{F}_{n-1}$ (see Fig.1). After 3D motion estimation, the information need to be recorded and transmitted include the indices of the corresponded vertex in $\widetilde{F}_{n-1}$ and the displacement (residuals) between the transformed current vertex and the corresponded vertex, written as (*v_index, x_residue , y_ residue , z_ residue*). After k-means clustering and ICP (i.e., 3D motion estimation), the vertices between two consecutive mesh models will be closer so that we can get much less residuals for encoding.

To further improve the coding efficiency, we introduce two procedures: "spatio-temporal search" and "local index search". Spatio-temporal search means that vertex prediction source can not only be from $\widetilde{F}_{n-1}$, but also from a subset of $\widetilde{F}_n$ itself. The subset that meets this purpose is limited to those vertices preceding the current

vertex in the decoding procedure (i.e., vertices with smaller indices). Those vertices not encoded yet will be excluded from consideration in spatio-temporal search. The search results in the spatial and temporal domains are compared and the one with less coordinate residual is chosen as the final prediction source for encoding.

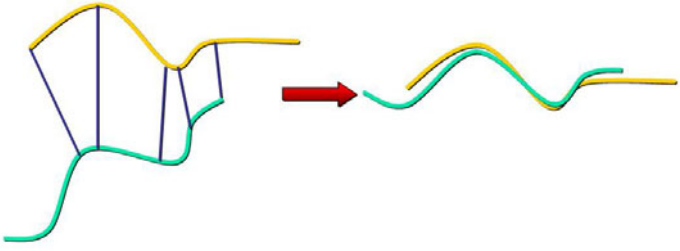


Fig. 1. ICP algorithm [2]

In addition to residuals of vertex coordinates, we still need to encode an extra list that records the indices of the corresponded vertices after 3D motion estimation. "Local index search" plus differential index coding will be beneficial to the coding efficiency of this list. In 3D motion estimation, we need to find a vertex in the previous model that is closest to the transformed current vertex. When the search range is restricted to locally neighboring indices of the prior encoded vertex, both the time complexity and the bit rate required for encoding the referred list can be significantly reduced. An exception is that a full search of $\tilde{\tilde{F}}_{n-1}$ will be still conducted if the residual of vertex coordinates from the local index search result is larger than a given threshold. The flow chart of our modified 3DMC algorithm based on inter-model prediction is summarized in Fig. 2.

4 Experiment Results

The 3D mesh models used in experiments are created from multi-view videos captured from 13 cameras arranged around the targeted objects, as shown in Fig. 3. For multi-view images at each time instant, the visual hull algorithm [3] is applied to reconstruct dynamic 3D mesh models. As mentioned earlier, dynamic 3D mesh models reconstructed in this way will not be time-consistent, that is, the number of vertices and the associated topology information will not be preserved. Frames 0-5 of "robot" (Fig.4) are used for experiments to test our proposed algorithm. (here, a "frame" means a reconstructed 3D mesh model at a time instant)

Table 1 shows the result of compression ratio. Since Frame-0 uses the MPEG-4 AFX-3DMC for encoding, we do not list it in Table 1. It is observed that the average compression ratio is 31.85 for Frames 1-5.

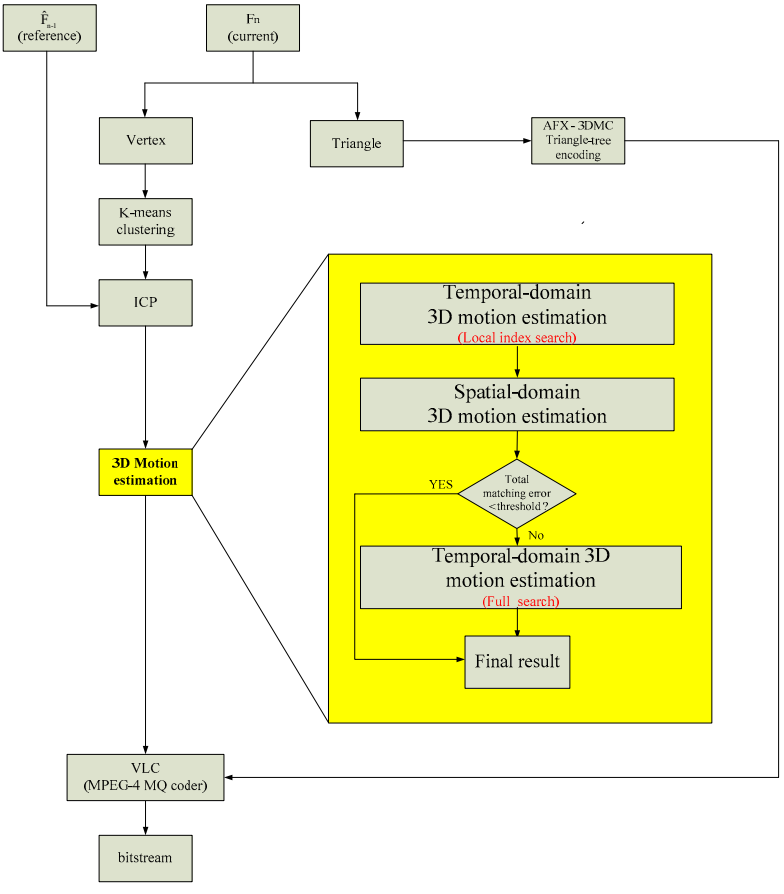


Fig. 2. The proposed modified 3DMC algorithm for time-inconsistent dynamic 3D mesh models

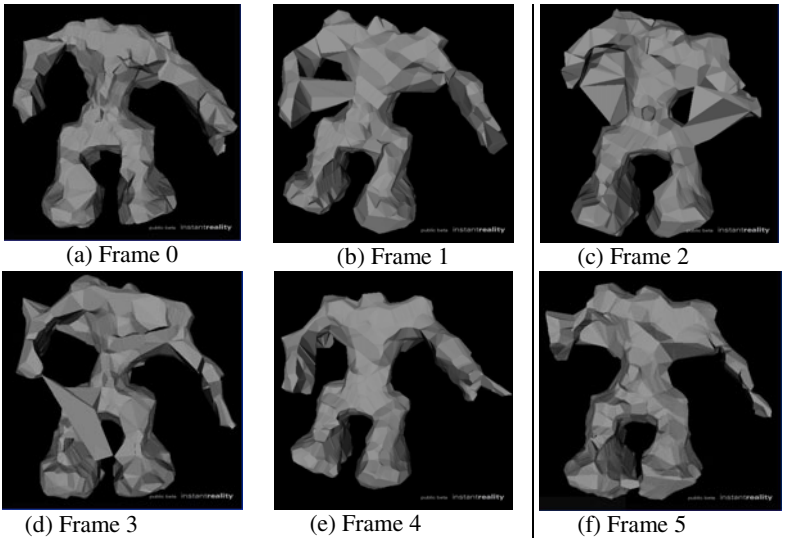
Fig. 3. Part of the multi-view video capturing configuration

Table 1. Compression ratio of our modified 3DMC algorithm

	Original file size (KB)	Compressed file size (KB)	Compression Ratio
Frame1	1339	42	31.88
Frame2	1074	35	30.69
Frame3	1278	40	31.95
Frame4	1323	41	32.27
Frame5	1154	37	31.19

Three quality measures are calculated for the decoded 3D mesh models:

(1) E_1: average norm-1 error (in terms of mm) of vertex position,
(2) E_2: KG error [9] (a well-known measurement in computer graphics),
(3) E_3: SNR (dB) of derived depth image (projecting the 3D mesh model onto a selected image plane to get depth image).



(a) Frame 0 (b) Frame 1 (c) Frame 2

(d) Frame 3 (e) Frame 4 (f) Frame 5

Fig. 4. 3D mesh models reconstructed based on multi-view images

To compare with the MPEG-4 AFX 3DMC, Figs. 5~7 show their R-D curves. Note that each data point of different bit rate is obtained by varying the coordinate accuracy (BPV=8,9,10,12,14 bit per vertex) for MPEG-4 AFX 3DMC or varying residual accuracy (Quality Factor, QF=1,2,5,10, 24, the larger, the more accuracy) for the modified 3DMC.

Figs. 5-7 show that our method outperforms MPEG-4 AFX 3DMC, regarding all three measures. At the same quality, our method has a compression gain of about 20% in bit rate.

We also conduct an experiment on computer-graphics-generated mesh model: "chicken" (Fig.8) to prove the applicability of our modified 3DMC on time-consistent models (but does not take advantage of the vertex correspondence relations). Similarly, we compute the three R-D curves for comparison (not shown here). At a considerable quality, our method outperforms MPEG-4 AFX 3DMC by a bit rate saving of 42.5% (4.6 KB/model vs. 8 KB/model). This better gain lies on the fact that a less noisy topology makes 3D motion estimation more reliable to finding matching vertices between two successive models.

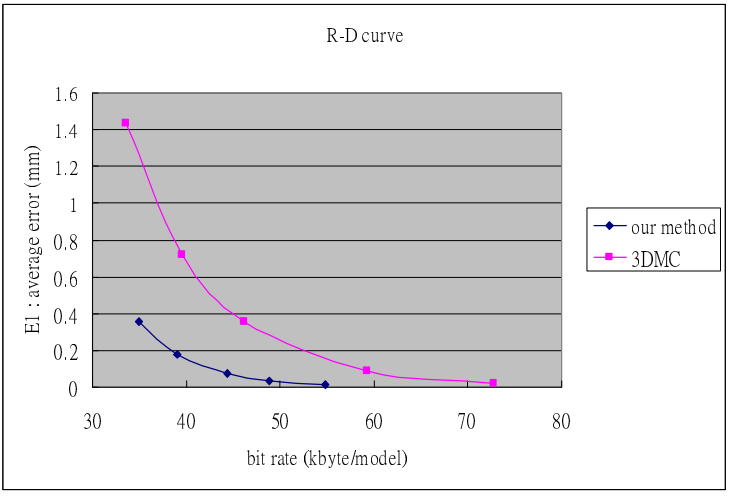


Fig. 5. The R-D curve (E_1 vs. bit rate) in comparison

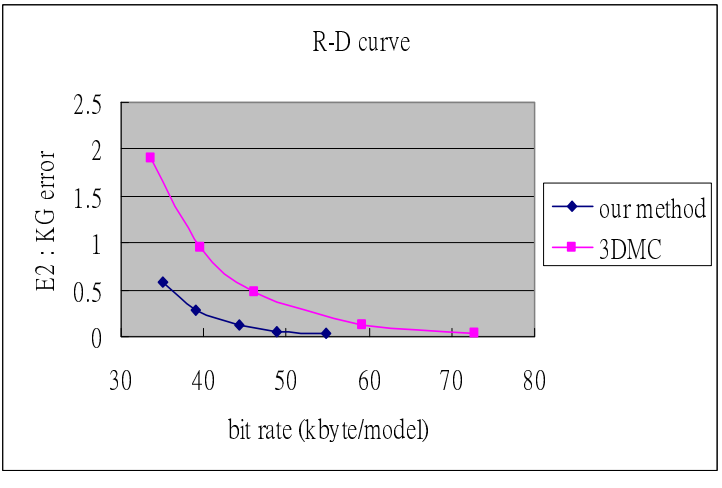


Fig. 6. The R-D curve (E_2 vs. bit rate) in comparison

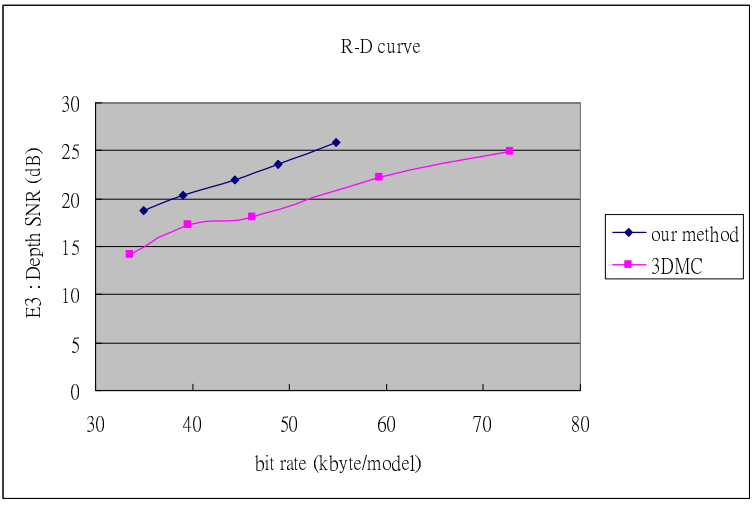


Fig. 7. The R-D curve (E_3 vs. bit rate) in comparison

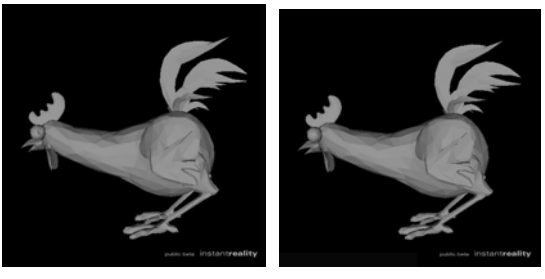


Fig. 8. Frames 0 & 1 of "Chicken" created by computer-graphics tools

5 Concluding Remarks

Essentially, our modified 3DMC algorithm is based on the traditional TS scheme, enhanced with a 3D motion estimation algorithm for vertex prediction between successive models. We also develop two algorithms of spatio-temporal search and local index search to further improve the coding efficiency. The compression ratio depends on the variation (e.g., global behavior or consistency of vertex motions) between two successive models, while that of 3DMC which compresses each model separately is kept less varying if the number of vertices is fixed. Another promising way to further improve coding efficiency is to build vertex correspondence and vertex ordering at the earlier stage of 3D mesh reconstruction, that is, adopting a preprocessing instead of a post-processing (via ICP algorithm).

References

1. Taubin, G., Rossignac, J.: Geometric Compression Through Topological Surgery. ACM Trans. on Graphics 17(2), 84–115 (1998)
2. Besl, P.J., McKay, N.D.: A Method for registeration of 3-D shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence 14(2), 239–256 (1992)
3. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Tran. on Pattern Analysis and Machine Intelligence 16(2), 150–162 (1994)
4. Merkle, P., Smolic, A., Muller, K., Wiegand, T.: Multi-View Video Plus Depth Respresentation and Coding. In: Proc. of IEEE Int'l. Conf. on Image Processing (ICIP 2007), vol. 1, pp. I-201 – I-204 (2007)
5. Laurentini, A.: The Visual Hull Concept for Silhouette-Based Image Understanding. IEEE Tran. on Pattern Analysis and Machine Intelligence 16(2), 150–162 (1994)
6. http://www.chiariglione.org/mpeg/technologies/mp-mv/
7. Stefanoski, N., Klie, P., Liu, X., Ostermann, J.: Layered Predictive Coding of Time-Consistent Dynamic 3D Meshes using a Non-Linear Predictor. In: Proc. of IEEE Int'l. Conf. on Image Processing (ICIP 2007), pp. V-109-V-112 (2007)
8. Amjoun, R., Sreaber, W.: Efficient Compression of 3D Dynamic Mesh Sequences. In: Proc. of International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, wscg (2007)
9. Karni, Z., Gotsman, C.: Compression of soft-body animation sequences. Computers & Graphics 28(1), 25–34 (2004)