PROGRESSIVE COMPRESSION OF 3D DYNAMIC MESH SEQUENCES

Jeong-Hyu Yang†, *Chang-Su Kim*‡ *and Sang-Uk Lee*†

†Signal Processing Lab., School of Electrical Engineering and Computer Science
Seoul National University, Seoul 151-741, Korea
‡Department of Information Engineering, The Chinese University of Hong Kong
Shatin, Hong Kong

ABSTRACT

An algorithm to compress 3D mesh sequences for dynamic objects is proposed in this work. Given an irregular mesh sequence, we construct a semi-regular mesh structure for the first frame and then map it to the subsequent frames based on the hierarchical motion estimation. The regular structure of the resulting mesh sequence enables us to employ the zero-tree coding scheme to compress the motion compensation residuals efficiently. Simulation results demonstrate that the proposed algorithm provides significantly better compression performance than the static coder that encodes each frame independently.

1. INTRODUCTION

Recently, it has drawn considerable attention to represent and compress static meshes for 3D objects efficiently. Among various 3D compression methods, the semi-regular mesh coding yields the state-of-the-art performance [1]. For 3D mesh sequences, several researchers have attempted to compress synthesized sequences [2]. However, relatively little effort has been made to compress real dynamic mesh sequences. In this work, we propose an algorithm to compress the mesh sequences for dynamic objects, which are captured with laser scanners.

Our work is related to the research on the mesh-based 2D video coding. To overcome the limitations of block-based motion models, mesh models have been proposed to represent content-adaptive and hierarchical motions [3, 4]. In these approaches, an initial mesh is constructed on the first frame, and the motion vector (MV) of each node point is estimated by using the optical flow method or the block matching algorithm. Then, the pixels within each mesh patch are motion-compensated by the spatial transformation, such as affine and bilinear transformations. MV's can be estimated by minimizing the energy function [3] or can be computed hierarchically in a coarse-to-fine order with connectivity constraints [4]. The proposed algorithm can be seen as the generalization of these approaches. However, while these approaches use 2D meshes to improve the motion compensated prediction in video coding, we are concerned with the representation and compression of the 3D shapes of dynamic objects.

Fig. 1 shows the block diagram of the proposed algorithm. We acquire an input mesh sequence with a laser scanner, which is initially given by a set of irregular mesh

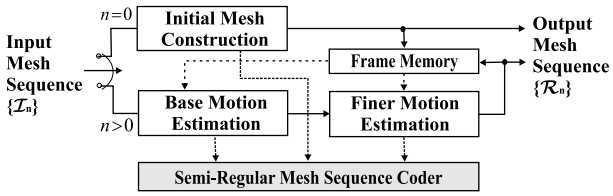


Fig. 1. Block diagram of the proposed algorithm.

frames. During the motion estimation, we convert the irregular mesh sequence into a semi-regular normal mesh sequence to achieve high coding efficiency. More specifically, the proposed algorithm constructs a semi-regular structure for the first frame, and then maps it to the subsequent frames based on the hierarchical motion estimation of base vertices and finer subdivision points. The hierarchical motion estimation guarantees that the prediction errors constitute the so-called zero-tree [1]. Therefore, these errors can be effectively compressed by the bit-plane coding scheme in a progressive way.

The paper is organized as follows. Section 2 describes the semi-regular mesh construction and the hierarchical motion estimation. Section 3 explains the compression method, and Section 4 presents simulation results. Finally, Section 5 concludes the paper.

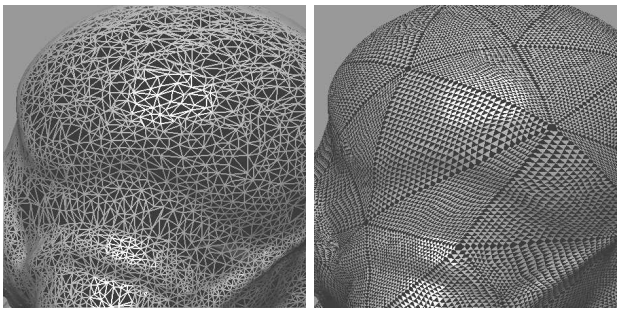
2. HIERARCHICAL MOTION ESTIMATION

Given an irregular mesh sequence $\{\mathcal{I}_n : n = 0, 1, 2, \ldots\}$, we construct a semi-regular normal mesh to describe the geometry of the first frame. Then, the motion estimation is conducted to track the movements of the node points of the first frame to the following frames. In this way, we can obtain the semi-regular mesh sequence, in which every frame has the same topology structure.

2.1. Initial Semi-Regular Mesh Construction

At this step, we convert the first frame $\mathcal{I}_0$ into a semi-regular normal mesh $\mathcal{R}_0$ [5]. The normal remeshing step is composed of the mesh simplification and the subdivision. In the mesh simplification, we obtain a base mesh of $\mathcal{I}_0$, $\mathcal{R}_0^0$, and the parameterization data of $\mathcal{I}_0$ onto $\mathcal{R}_0^0$. Then, the base mesh is refined to $\mathcal{R}_0^1$ by subdividing each triangle into four triangles. The subdivision points are computed

0-7803-8554-3/04/$20.00 ©2004 IEEE.



(a) Irregular mesh (b) Semi-regular Mesh

Fig. 2. Comparison of irregular mesh and semi-regular mesh. For the sake of illustration, the normals of triangles in the semi-regular mesh are flipped alternatively.

by the butterfly prediction and normal shooting scheme, which uses the parameterization data. This subdivision is iteratively applied L times to obtain the hierarchy

$$\mathcal{R}_0^0 \subset \mathcal{R}_0^1 \subset \cdots \subset \mathcal{R}_0^L = \mathcal{R}_0. \tag{1}$$

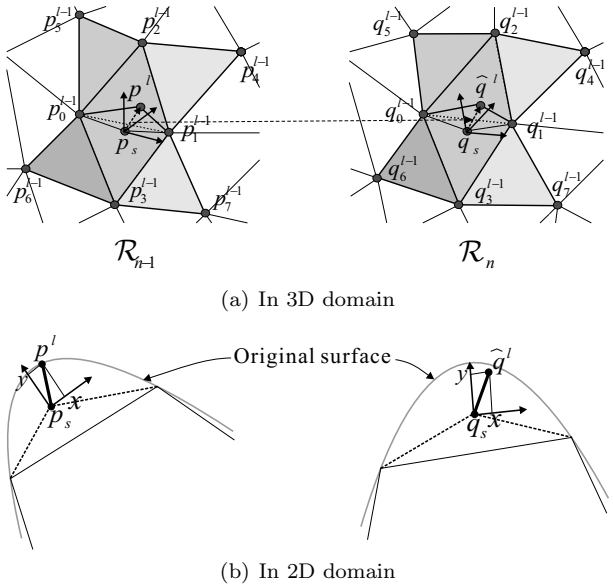
The geometry information in the hierarchy is represented by the base mesh points and the wavelet coefficients. Therefore, the geometry of the semi-regular mesh $\mathcal{R}_0$ can be effectively compressed by adopting wavelet coding schemes [1]. In Fig. 2, we can compare the regularity of the semi-regular mesh with that of the original irregular mesh.

2.2. Base Motion Estimation

Given the previous semi-regular mesh $\mathcal{R}_{n-1}$, we construct the base mesh $\mathcal{R}_n^0$ of the current frame by estimating the motions from $\mathcal{R}_{n-1}^0$ to the geometry domain of $\mathcal{I}_n$. We attempt to solve this correspondence problem in a similar way to the region-based motion estimation [6], which was introduced to explain the motions of different objects in 2D video sequences. The image segmentation and the motion estimation of each segment are jointly performed, such that the motion of each segment is represented accurately with a parametric model.

We extend this idea to 3D motion estimation. Beginning with the whole base mesh $\mathcal{R}_{n-1}^0$ as an initial segment, we compute its motion parameter. The motion parameter of a segment is composed of the rotation matrix and the translation vector, which is computed by a modified version of the ICP algorithm [7]. The outliers, which cannot be represented well with the motion parameter, constitute new segments. Since this procedure can break the connectivity, we postprocess the outliers through a merging-splitting-merging procedure to obtain connected segments. For each segment, we repeat the motion estimation and the splitting until the stopping criteria are satisfied. In the motion estimation of each segment, we constrain the motion search region, since flexible object motions make it difficult to match correspondences by finding the closest points in the whole region.

The joint mesh segmentation and motion estimation with the ICP algorithm can describe only piecewise rigid motions. Thus, the motion of each vertex of the base mesh



(a) In 3D domain

(b) In 2D domain

Fig. 3. Motion interpolation.

should be refined to compensate flexible motions. A point $\mathbf{p}$ in $\mathcal{R}_{n-1}^0$ is moved to $\bar{\mathbf{q}}$ by its motion parameter, and then is mapped onto the closest point $\mathbf{q}$ within $\mathcal{I}_n$. The position of $\mathbf{q}$ is finally refined by moving it along neighboring vertices in $\mathcal{I}_n$. We perform the motion refinement by minimizing the following energy function

$$E(\mathcal{R}_n^0) = \sum_{\mathbf{q}\in\mathcal{R}_n^0} \{w_s E_s(\mathbf{q}) + w_m E_m(\mathbf{q})\}, \tag{2}$$

where w_s and w_m are weighting coefficients. The first term $E_s(\mathbf{q})$ represents the temporal change in the spring energy of the edges which are incident on $\mathbf{q}$. The second term $E_m(\mathbf{q})$ is introduced to preserve the surface shapes, characterized by normal vectors and curvatures. The energy function in (2) is iteratively minimized by choosing the neighboring vertex which yields the maximum energy reduction.

2.3. Finer Motion Estimation

Given $\mathcal{R}_{n-1}$ and $\mathcal{R}_n^0$, the MV of a subdivision point $\mathbf{q}^l$ in $\mathcal{R}_n^l$ $(l = 1, \ldots, L)$ can be estimated using the MV's of points in $\mathcal{R}_n^{l-1}$. Note that the position of $\mathbf{q}^l$ is not yet determined, but is to be computed at this step. Let $\hat{\mathbf{q}}^l$ denote the predicted point of $\mathbf{q}^l$. Then, as shown in Fig. 3(a), $\hat{\mathbf{q}}^l$ is given by

$$\begin{aligned} \hat{\mathbf{q}}^l &= \sum_{i=0}^{7} w_i \mathbf{q}_i^{l-1} + C(\mathbf{p}^l - \sum_{i=0}^{7} w_i \mathbf{p}_i^{l-1}), \\ &= \mathbf{q}_s + C(\mathbf{p}^l - \mathbf{p}_s), \end{aligned} \tag{3}$$

where w_i is the ith butterfly filter coefficient. $\mathbf{p}_i^{l-1} \in \mathcal{R}_{n-1}^{l-1}$ and $\mathbf{q}_i^{l-1} \in \mathcal{R}_n^{l-1}$ are, respectively, the ith butterfly points for $\mathbf{p}^l$ and $\mathbf{q}^l$. $C(\cdot)$ is the local coordinate transform. The local coordinates are computed such that $\mathbf{p}_s$ and $\mathbf{q}_s$ become

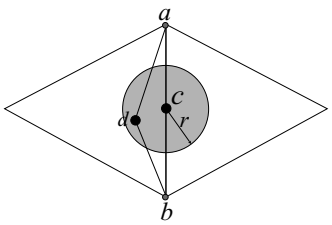


Fig. 4. A validity check of normal shooting.

the origins and z-axes are in the surface normal direction at $\mathbf{p}^s$ and $\mathbf{q}^s$. Also, the tangential axes are defined in a similar way.

In (3), $\mathbf{p}^l - \mathbf{p}_s$ is a spatial wavelet coefficient in the previous frame. Thus, $\hat{\mathbf{q}}^l$ is compensated by the wavelet coefficient in the previous frame. As shown in Fig. 3(b), $\mathbf{p}^l$ and $\hat{\mathbf{q}}^l$ are located in the same position in each local coordinates and we have

$$\hat{\mathbf{q}}^l - \mathbf{q}_s = C(\mathbf{p}^l - \mathbf{p}_s).$$

Therefore, the coordinate transform $C(\cdot)$ in (3) guarantees that $\hat{\mathbf{q}}^l$ becomes equal to $\mathbf{q}^l$, provided that the object moves rigidly and the base mesh is estimated accurately.

The finer motion estimation of $\mathbf{q}^l$ is completed by mapping $\hat{\mathbf{q}}^l$ onto the surface of $\mathcal{I}_n$. We find the piercing point in the normal direction from $\hat{\mathbf{q}}^l$. The validity of this normal shooting is checked in the parameterization domain to avoid undesirable artifacts. To this end, we compute the parameterization information from $\mathcal{I}_n$ onto $\mathcal{R}_n^0$ [8]. Specifically, we find the pathes corresponding to the edges in $\mathcal{R}_n^0$, group the vertices enclosed by the three pathes of each triangle, and obtain the parameterization data by computing the harmonic map within each triangle. As the triangles are subdivided, the parameterization data are rearranged. Then, the validity of the mapping is checked in the 2D parameter domain as shown in Fig. 4, where the $\mathbf{c}$ is the parametric candidate point at the center of the edge $\overline{\mathbf{ab}}$ and $\mathbf{d}$ is the candidate point obtained by the normal shooting. If $\mathbf{d}$ is within the circle, the normal shooting is declared as valid and the inverse mapping of $\mathbf{d}$ becomes the final position $\mathbf{q}^l$. Otherwise, the inverse mapping of $\mathbf{c}$ is employed as $\mathbf{q}^l$.

Finally, the motion prediction error $\mathbf{e}$ of $\mathbf{q}^l$ is given by

$$\mathbf{e} = \mathbf{q}^l - \hat{\mathbf{q}}^l, \tag{4}$$

which is encoded by the following coding scheme.

3. SEMI-REGULAR MESH SEQUENCE CODING

We propose a coding scheme which compresses the semi-regular normal mesh sequences, which are converted from irregular input meshes. The base mesh of a semi-regular mesh is the starting structure from which the subdivision is recursively performed. Its connectivity is typically irregular, while subdivision points form a hierarchical regular structure. Therefore, the semi-regular mesh sequence is compressed in two phases: irregular base mesh sequence coding and subdivision geometry sequence coding.

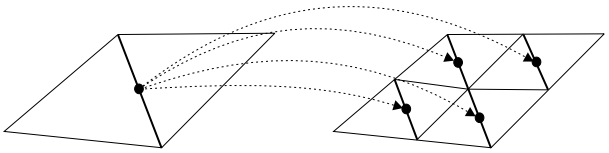


Fig. 5. Parent-child relations among subdivision points.

3.1. Coding of Base Mesh Sequence

Since the semi-regular mesh sequence has the unified topology due to the motion tracking, its base mesh sequence also has the isomorphic property. In other words, the number of vertices and the topology are invariant over all the frames. Yang *et al.* proposed a compression algorithm for 3D isomorphic mesh sequences based on the two-stage vertex-wise MV prediction [9]. At the first stage, the MV of a vertex is predicted from the weighted sum of the neighboring MV's. At the second stage, the remaining redundancies are further removed by exploiting the spatial or temporal correlation adaptively. We apply this algorithm to compress the base mesh sequence embedded in the semi-regular mesh sequence. In this algorithm, the prediction errors are uniformly quantized and arithmetic-coded. Note that there still remain the quantization errors, which are reduced further in the zero-tree coding in the next phase.

3.2. Coding of Subdivision Geometry Sequence

The zero-tree coding scheme was successfully applied to subdivision geometry shape [1], using the parent-child relations in the semi-regular mesh, which is shown in Fig. 5. After encoding the first frame by the conventional algorithm [1], the prediction error vectors $\mathbf{e}$ in (4) are compressed by the zero-tree coding scheme SPIHT [10]. The zero-tree coder consists of three independent coders: two for tangential components and one for the normal component of the prediction errors. The bitstreams from these three coders are interleaved to achieve progressiveness.

The bit allocation for each frame affects coding efficiency considerably. In this work, we apply common bit planes for all prediction errors in a sequence, and encode the same level of bit planes of all prediction errors sequentially. As bits are received, the total energy of prediction errors decrease monotonically. During the zero-tree coding, the quantization errors of the base mesh sequence are included at the root nodes. Then, the output bits of the zero-tree coder are entropy-coded. In the decoder, $\hat{\mathbf{q}}^l$ in (3) is computed with the decoded values of points. Finally, when all bit planes are received, the decoded meshes become the same with the original semi-regular meshes.

4. SIMULATION RESULTS

To evaluate the proposed algorithm, we made a clay model called 'dancer', which has bones and can be moved in a human-like way. We animated the model by hands, and scanned each frame with the Cyberware 3030MS scanner. The test sequence consists of five frames, and each frame is composed of 42,990 vertices and 85,976 triangles on average.

(a) 1st frame (b) 5th frame

Fig. 6. The 'dancer' test sequence.

Fig. 6 shows the first and the fifth frame meshes. The first frame is simplified to a base mesh which has 177 vertices and 350 triangles. Then, the base mesh is recursively subdivided four times to make a semi-regular normal mesh composed of $44,802$ vertices and $89,600$ triangles. The following frame meshes are converted into semi-regular normal meshes by the base motion estimation and the finer motion estimation.

Fig. 7 presents the compression results. The distortion is measured by the METRO algorithm [11], which computes the dissimilarity between two arbitrary meshes. The distance is normalized by the bounding box diagonal and thus has no unit. For the sake of comparison, we convert each frame mesh into a semi-regular normal mesh and compress the resulting mesh independently without considering temporal correlations. This scheme is called 'intra-coding' in Fig. 7. On the contrary, the proposed algorithm is called 'inter-coding,' since it yields a significant coding gain by exploiting the temporal correlations. When the target distortion is fixed, the proposed algorithm requires only about half the bit rate of the intra-coding scheme. For example, when the distortion is 0.0006, the proposed algorithm produces about 1.3 Kbytes/frame whereas the intra-coding scheme requires about 2.3 Kbytes/frame. In this test, we do not include the bits for the base mesh sequences to compare only the performances of subdivision geometry compression. When they are included, the proposed algorithm provides even better coding gain.

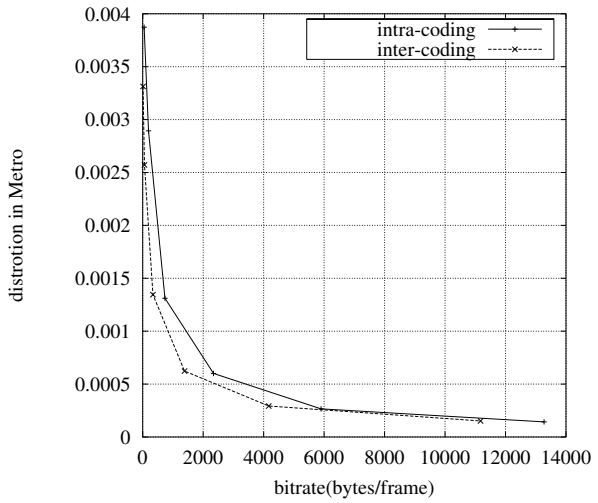


Fig. 7. The comparison of compression performances. The 'intra-coding' denotes the independent coding of each frame, whereas the 'inter-coding' denotes the proposed algorithm based on the hierarchical motion estimation.

5. CONCLUSION

In this paper, we proposed an algorithm to compress 3D dynamic mesh sequences. The proposed algorithm first converts an input irregular mesh into the semi-regular one based on the hierarchical motion estimation. The semi-regular mesh sequence has a highly regular structure, which enables us to apply the zero-tree coder to compress the residual vectors effectively. Simulation results demonstrated that the proposed algorithm shows much better compression performance than the intra-coding scheme that encodes each frame independently.

6. REFERENCES

[1] A. Khodakovsky, and I. Guskov, "Normal mesh compression," in *Geometric modeling for scientific visualization*, Springer-Verlag, Heidelberg, Germany, 2002.

[2] S. Gupta, K. Sengupta, and A. Kassim, "Registration and partitioning-based compression of 3-D dynamic data ," *IEEE Trans. CSVT*, vol. 13, no. 11, Nov. 2003.

[3] Y. Wang, and O. Lee, "Active mesh-A feature seeking and tracking image sequence representation scheme," *IEEE Trans. IP*, vol. 3, pp. 610–624, Sept. 1999.

[4] P. V. Beek, A. M. Tekalp, N. Zhuang, I. Celasun, and M. Xia, "Hierarchical 2-D mesh representation, tracking, and compression for object-based video," *IEEE Trans. CSVT*, vol. 9, no. 2, March 1999.

[5] I. Guskov, K. Vidimce, W. Sweldens, and P. Schroder, "Normal meshes," in *SIGGRAPH 2000*, pp. 95–102, 2000.

[6] Y. Wang, J. Ostermann, and Y.-Q. Zhang, *Video processing and communications*, Prentice Hall, 2002.

[7] P. Besl and N. McKay, "A method for registration of 3-D Shapes," *IEEE Trans. PAMI*, vol. 14, no. 2, 1992.

[8] J.-H. Yang, C.-S. Kim, and S.-U. Lee, "Robust semi-regular mesh representation of 3D dynamic objects," in *ISCAS'03*, pp. II-664–II-667, May. 2003.

[9] J.-H. Yang, C.-S. Kim, and S.-U. Lee, "Compression of 3D triangle mesh sequences based on vertex-wise motion vector prediction," *IEEE Trans. CSVT*, vol. 12, no. 12, Dec. 2002.

[10] A. Said, and W. A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. CSVT*, vol. 6, pp. 243–250, June 1996.

[11] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," in *Computer Graphics Forum*, 17, 2, pp. 167–174, 1998.