Generation of Fixed Spectral Basis for Three-Dimensional Mesh Coding Using Dual Graph

Sung-Yeol Kim, Seung-Uk Yoon and Yo-Sung Ho
Gwangju Institute of Science and Technology (GIST)
Gwangju, 500-712, **Korea**
Tel: +82-062-970-2263 E-mail: {sykim75, suyoon, hoyo}@gist.ac.kr

Abstract:

In this paper, we propose a new scheme for geometry coding of three-dimensional (3-D) mesh models using a fixed spectral basis. In order to code the mesh geometry information, we generate a fixed spectral basis using the dual graph derived from the 3-D mesh topology. After we partition a 3-D mesh model into several independent submeshes to reduce coding complexity, the mesh geometry information is projected onto the generated orthonormal bases which are the eigenvectors of the Laplacian matrix of the 3-D mesh. Finally, spectral coefficients are coded by a quantizer and a variable length coder. The proposed scheme can not only overcome difficulty of generating a fixed spectral basis, but also reduce coding complexity. Moreover, we can provide an efficient multi-resolution representation of 3-D meshes.

Keywords: Fixed spectral basis, Dual graph, Mesh coding

1. INTRODUCTION

As demands for high quality visual services have increased from consumers and the interest of three-dimensional (3-D) meshes has grown rapidly, it is essential to develop efficient 3-D mesh data coding methods. As one of 3-D representation methods, the mesh model is simply a set of planar polygons in the 3-D Euclidean space. In order to represent a mesh surface, we assume that a 3-D model consists of triangular faces.

Fundamentally, there are three types of information to describe the mesh surfaces; geometry, connectivity, and photometry information. The geometry information describes 3-D coordinates of mesh vertices, and the connectivity information describes the topology with the incidence relations among vertices, edges and faces. The photometry information includes surface normal vectors, colors, texture coordinates that are the attributes of geometry information. These three data sets can be the targets for 3-D mesh coding. In this paper, we focus on the mesh geometry information.

Mesh geometry coding methods can be divided into two categories: spatial prediction methods and spectral methods. Deering [1] and Taubin et al. [2] ordered the vertices according to the connectivity information, and then coded the vertices using a simple linear predictor in the spatial domain. Similarly, the mesh coding scheme of Touma and Gotsman [3] developed an algorithm to code the topology as a traversal of the vertices in the spatial domain, and the vertex coordinates are coded by predicting them along the traversal order using a parallelogram scheme. Finally, the prediction errors are entropy-coded.

On the other hand, Karni and Gotsman [4] proposed the spectral methods for the 3-D mesh geometry coding. Karni and Gotsman projected the mesh geometry onto basis vectors which are the eigenvectors of the mesh Laplacian matrix. Although Karni and Gotsman obtain good results from the spectral method, there are some critical problems such as the difficulty of fixed spectral basis generation and the tremendous coding complexity.

In this paper, we are concerned about the mesh geometry coding using the spectral method and try to solve the fixed spectral basis generation problem. We propose a new scheme for the orthonormal spectral basis generation. Moreover, we partition 3-D mesh models into several independent pieces to reduce coding complexity.

This paper is organized as follows. After we review the previous works in Section 2, Section 3 describes the proposed spectral coding method of the mesh geometry. Section 4 provides experimental results, and we make conclusions in Section 5.

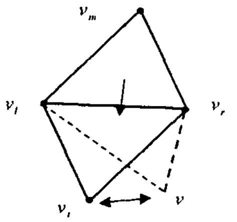
2. MESH GEOMTRY CODING

2.1 Spatial Prediction Methods

The parallelogram prediction is the most famous coding scheme for mesh geometry. Touma and Gotman [3] proposed the parallelogram prediction method in 1998. Basically, the parallelogram prediction can be one of spatial prediction methods since it is performed in the 3-D spatial domain.

We can get the prediction vertex $v_t=(x_t,y_t,z_t)$ of each vertex $v=(x,y,z)$ based on the parallelogram prediction in the triangle spanning tree as we can see in Fig.1.

Before we apply the parallelogram prediction, we first traverse all vertices to obtain the coding order that is called by triangle tree traversal. Then, we coded the vertex coordinates along the triangle traversal.



$$v_t = v_l + v_r - v_m$$

Fig. 1. Parallelogram prediction

For example, when we pass the triangle 1 to triangle 2, the vertex v_m, v_l, and v_r are already coded. The opposite vertex v from the common edge (v_m, v_l) is predicted as v_l+ v_r-v_m. As a result, the predicted vertex v_t forms a parallelogram and belong the same plane with the three ancestors. Finally, we calculate and code the prediction errors by v – v_t. For some vertices, all three ancestors may not be available. When we use two ancestors, the prediction coefficients are set to 2 and –1. If there is only one ancestor, we use the ancestor directly as the prediction value. In case of no ancestor, a null prediction is used.

2.2 Spectral Methods

Karni and Gotsman show us how to extend the classical Fourier analysis to 3-D mesh data [4]. We can assume a simple 3-D mesh model which is composed of n vertices. The adjacency matrix A of the model can be represented with the circular n x n matrix. The diagonal matrix D also can be represented with the n x n matrix. Finally, we can obtain the so-called Laplacian matrix from the adjacent matrix A and the diagonal matrix D. The Laplacian matrix
describes the analog of the second spatial derivative conceptually.

The Fourier basis functions for 2-D signals are obtained as the eigenvectors of the Laplacian matrix of the graph with the topology of a 2-D grid. Karni and Gotsman try to adopt the 2-D spectral transformation to the 3-D mesh topologies. First, we can define the adjacency matrix A and diagonal matrix D.

$$A_{ij} = \begin{cases} 1 & \text{i and j are neighbors} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

$$D_{ij} = \begin{cases} \mathrm{d_i} & \text{i = j, valence of i} \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

The diagonal components of D describe the valence of each vertex. The Laplacian matrix L can be obtained using the equation such that L = I - DA, where I is the identity matrix

$$L_{ij} = \begin{cases} 1 & i = j \\ -1/d_i & \text{i and j are neighbors} \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Karni and Gotsman perform the spectral coding by projecting the geometry data onto the eigenvectors of Laplacian matrix L and then the spectral coefficients are quantized. We should note that the eigenvectors may not be fixed since the valences of vertices are not fixed. In other words, the spectral bases are different according to the mesh topology. Karni and Gotsman try to solve the problem by mapping the arbitrary mesh topology into a regular mesh topology [5]. However, serious deformation has occurred when the arbitrary meshes are mapped into a regular mesh as we can see in Fig. 2. In this paper, we propose a new algorithm to generate the fixed spectral basis.

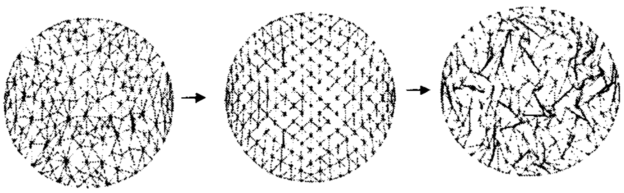


Fig. 2. Deformation problem

3. SPECTRAL GEOMTRY CODING

3.1. Spectral Coding Method Using Dual Graph

In ordèr to code the 3-D mesh geometry, we propose a spectral coding using a fixed spectral basis. The proposed method calculates a fixed spectral basis from the dual graph derived from the mesh topology. As we mentioned in the spectral mesh geometry coding, it is difficult to generate a fixed basis since the topology is different according to the mesh model. We try to extend previous 2-D transform coding based on a fixed spectral basis into 3-D mesh coding.

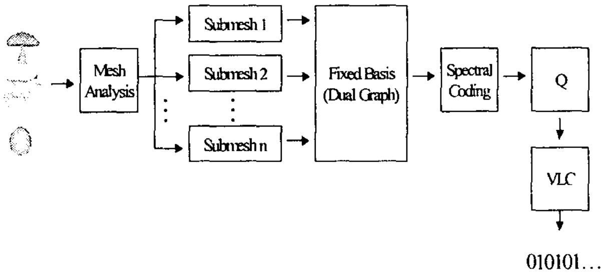


Fig. 3. Block diagram of spectral geometry coding

Fig. 3 shows the entire block diagram for the proposed system. First, we analyze the input mesh models into the three types information. In other words, we extract the geometry, connectivity and photometry information from the input model. Then, we partition the mesh into several independent pieces to reduce complexity using the extracted connectivity information. We call the independent pieces as a submesh. During a mesh partitioning, we obtain the dual graph for each submesh. After that, we find the fixed spectral basis from the property of the dual graph. Finally, the mesh geometry is projected onto the generated fixed spectral basis and the transformed geometry data are coded through a quantizer and a variable length coder.

3. 2 Mesh Partitioning

Mesh partitioning techniques are used to divide a given 3-D mesh model into several independent pieces. We call an independent piece as a submesh. When submeshes are transmitted instead of the entire model, it is useful for storage and transmission since spatial transmission errors will not affect the entire model at the receiver side. The main reason that we carry out the mesh partitioning in this paper is to decrease computational complexity.

We apply a multi-seed traversal technique that is one of the well-known partitioning techniques [6]. One of the important parts in mesh partitioning is to select the initial vertex positions that are the starting points of the partitioning process. In this paper, we use the K-means clustering algorithm to select initial vertices [7]. First, we set the number of the initial submeshes *K* that is the total number of child submeshes obtained from a parent submesh. The K-means algorithm performs to find positions of initial vertices as many as *K*. The initial vertices are selected by.

1. Choose *K* initial center vertices $z_1(1)$, $z_2(1)$,...,$z_K(1)$ by selecting arbitrary *K* vertices in the parent submesh.
2. At the k-th iterative step, distribute the input vertices {X} among the *K* cluster domains by eq. 4

$$x \in S_j(k) \quad if \quad \| x - z_j(k) \| < \| x - z_i(k) \| \qquad (4)$$

for all $i = 1, 2, ..., K$, i j , where $S_j(k)$ indicates the set of vertices whose center is $z_j(k)$.

3. From Step 2, compute new cluster centers $z_j(k+1)$, j = 1,2,...*K*, such that the sum of squared distances from all points in $S_j(k)$ to the new cluster center is minimized. In other words, the new cluster center $z_j(k+1)$ is computed to minimize the performance index

$$J_j = \sum_{x \in S_j(k)} \| x - x_j(k+1) \|^2, \quad j = 1, 2, ..., K \qquad (5)$$

Here, we note that $z_j(k+1)$ is simply the sample mean of $S_j(k)$. Therefore, the new vertices center is given by eq. 6

$$z_j(k+1) = \frac{1}{N_j} \sum_{x \in S_j(k)} x \qquad \mathrm{j} = 1, 2, ..., K \qquad (6)$$

where N_j is the number of samples in $S_j(k)$.

4. If $z_j(k+1) = z_j(k)$ for j = 1,2,...,K, the algorithm has converged and the procedure will be terminated. Otherwise, go to Step 2. If the procedure is terminated, the new center vertices $z_1(k+1)$, $z_2(k+1)$,..., $z_K(k+1)$ become initial vertices.

As a result, the center position is convergent to new vertex that is the optimal position of the initial vertex. Therefore, the mesh partitioning from these initial vertices can be operated in an optimal manner. On the contrary, the K-means clustering algorithm needs more computational time than other algorithms, such a maximum-distance algorithm. However, we do not need the real-time partitioning techniques since the submeshes may be stored in a data storage through a off-line process in advance.

Fig. 4(a) and Fig. 4(b) show selections of initial vertices by the maximum-distance algorithm and the K-means algorithm, respectively, when the number K of the initial submeshes is three. In Fig. 4, we can notice that the initial vertex positions selected by the K-means algorithm are more optimal than those selected by the maximum distance algorithm.

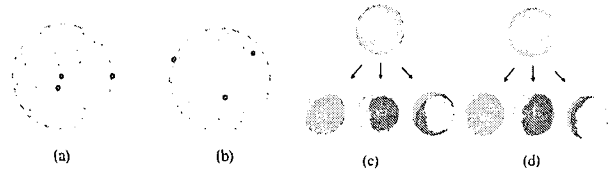


Fig. 4. Initial vertex selection (a) maximum distance algorithm (b) K-means algorithm (c) the result of (a), (d) the result of (b)

3.3 Generation of Fixed Basis

Given a planar graph G, its geometric dual graph G* is constructed by placing a vertex in each region of G. Fig. 5 shows an example of dual graph for simple planar graph. The dual graph G* of a polyhedral graph G has dual graph vertices each of which corresponds to a face of G and each of whose faces corresponds to a graph vertex of G.

In this paper, we use the dual graph G* for generating a fixed spectral basis. After obtaining submeshes from mesh partitioning, we generate their dual graphes by connecting the centers of gravity of triangle faces for each submesh. One of important concerns for a dual graph is whether we can regenerate the mesh geometry from its dual graph.

As we can see in Fig. 6, we should know the first two vertices of each submesh in order to regenerate the mesh geometry. We can make a triangle face from the two vertices and the one vertex of dual graph. Basically, we can define a plane with three vertices: the first two vertices v1 and v2 and the one vertex of dual graph d1. The third vertex v3 of a triangle face is located on the defined plane. We calculate the midpoint t1 from v1 and v2. The third vertex v3 is located on the line from d1 and t1. Finally, we can find the third vertex v3 by advancing two times of the distance between d1 and t1 from d1.

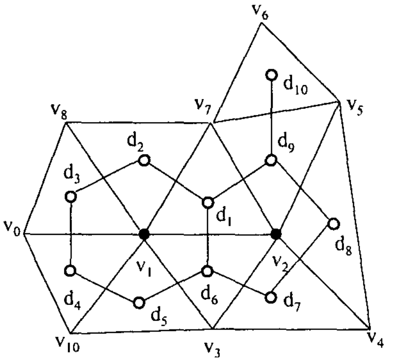


Fig. 5. Polyhedral graph and its dual graph

With this kind of procedure, we can regenerate the mesh geometry perfectly. In other words, we can replace the mesh geometry with the first two vertices of mesh geometry and its dual graph. Therefore, we compress these information instead of the mesh geometry.

Since the mesh models are composed of triangular faces, most of the valences of its dual graph vertices are 3. This property is the idea to generate a fixed spectral basis. The main problem of creating a fixed spectral basis is the irregularity of the mesh. We can solve the problem by using the duality between the mesh topology and its dual graph.

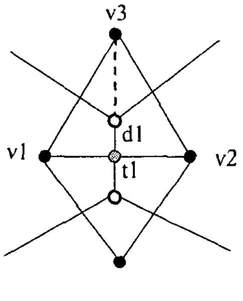


Fig. 6. Regeneration of mesh geometry with its dual graph

We can notice that the valence of dual graph vertex is 3 when the triangle face is the inner face in the mesh in Fig. 5. When the triangle face is on the boundary of the mesh and is not an ear, the valence of dual graph vertex is 2. In case the triangle face is an ear, we can assume the valence of dual graph vertex is 1.

With these assumptions, we can generate the fixed spectral basis by analyzing the each submesh topology. Basically, since the topology information is coded and transmitted prior to the mesh geometry, we can extract the boundary information at the decoder side. As a result, we can make a basis for the spectral coding according to the boundary information and the number of inner triangle.

3.4 Spectral Coding of Mesh Geometry

After the dual graph vertices are projected onto the generated basis, the spectral coefficients are coded through a quantizer and a variable length coder. First, the spectral coefficients are uniformly quantized with the specified level that can be between 10 and 16 bits. The reduction of geometry information is actually achieved by the quantization step. Finally, the quantized coefficients are entropy coded using a Huffman or arithmetic coder. We can provide the progressive transmission and multiresolution representation of 3-D meshes by selecting the spectral coefficients to be sent.

3.5 Hausdroff Distance

In order to evaluate the distortion between the original 3-D meshes and the reconstructed ones, we should define a error metric. For the 3-D error metric, Hausdroff distance is widely used. The original and reconstructed models have same topology information. However, they may have different values of vertex positions. The closest vertex of a reconstructed model is selected and the distance is calculated when we define the distance by eq. 7.

$$d(p,S') = \min_{p' \in S'} \|p - p'\| \qquad (7)$$

Here, d(p,S') denotes the distance between p vertex and surface S'. As a result, Hausdroff distance can be defined by eq. 8.

$$d(S,S') = \max_{p \in S} d(p,S') \qquad (8)$$

Here, S denotes the original model and S' denotes the reconstructed model respectively.

4. EXPERIMENTAL RESULTS

We have evaluated the proposed algorithm with the COW model. It consists of 2903 vertices and 5804 faces. We partitioned the COW model into 20 submeshes. We also used Hausdroff distance, which is the common measure of 3-D model deformation, to compare the proposed algorithm with the previous algorithm.

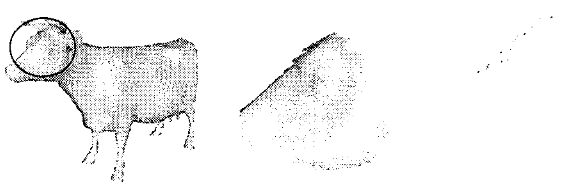


Fig. 7. 19th submesh of the COW model

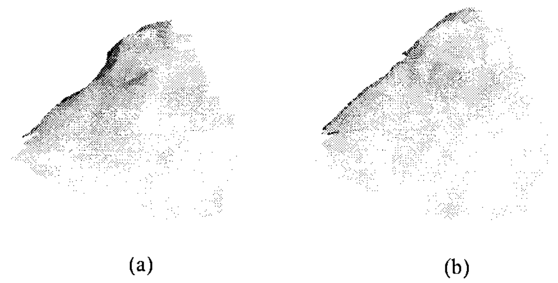


Fig. 8. Result of the 19th submesh (a) basis generation using a regular mesh (b) basis generation using a dual graph

Fig. 7 shows the 19th submesh of the COW model and Fig. 9 shows the result of spectral coding. We assume that the decoder know the the number of inner vertices, boundary and ear of the 19th submesh. We generated 196 fixed spectral bases for the 19th submesh. For the experiment, we code 150 spectral coefficients.

As you can see in Fig. 8, we can notice some deformation in both algorithms. However, our proposed algorithm had less deformation than the previous algorithm in the left eye region of COW model. The main reason is that there is mismatching between the original mesh and a regular mesh in the previous algorithm since the left eye region of COW region is not regular. Table 1 also shows the Hausdroff distance is smaller in the proposed system.

Fig. 9. 4th submesh of the COW model

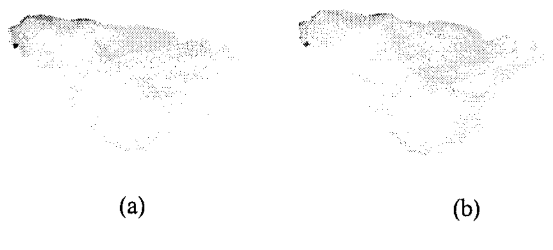


Fig. 10. Result of the 4th submesh (a) basis generation using a regular mesh (b) basis generation using a dual graph

Fig. 9 also shows the 4th submesh of the COW model and Fig. 10 shows the results of the 4th submesh. As we can see in Fig. 9, the valances of the 4th submesh vertices are all 6. As a result, the previous algorithm works better than ours. In other words, the algorithm using a regular mesh can be performed well than our algorithm. As we can imagine, our algorithm have error propagation problem when we regenerate the mesh geometry from the dual graph.

One of method to solve the error propagation problem is the insertion of a real mesh geometry information in encoder side to refresh the propagated error. We also develop an algorithm to overcome the propagation problem by selecting mesh geometry coding mode adaptively.

Table 1. Comparison of Hausdroff distance

Submesh	Vertices	Coeff.	RegularMesh	DualGraph
1st	182	130	0.0055	0.0059
2nd	156	130	0.0079	0.0062
3rd	158	130	0.0069	0.0059
4th	186	130	0.0104	0.0118
5th	187	150	0.0072	0.0068
6th	182	150	0.0058	0.0051
7th	152	130	0.0046	0.0049
8th	177	130	0.0102	0.0093
9th	174	150	0.0046	0.0039
10th	159	130	0.0064	0.0048
Submesh	Vertices	Coeff.	RegularMesh	DualGraph
11th	185	150	0.0088	0.0075
12th	159	130	0.0041	0.0036
13th	172	150	0.0047	0.0039
14th	198	150	0.0072	0.0071
15th	188	150	0.0042	0.0049
16th	164	130	0.0051	0.0045
17th	161	130	0.0027	0.0031
18th	174	130	0.0045	0.0041
19th	170	150	0.0058	0.0035
20th	173	150	0.0069	0.0052

5. CONCLUSIONS

In this paper, we proposed a new algorithm for mesh geometry spectral coding. We generate the fixed spectral basis from the dual graph derived from the mesh topology. After we generate fixed spectral basis, we project the mesh geometry onto the basis. Then, we code the spectral coefficients with a quantizer and a variable length coder. The proposed algorithm can reduce the deformation after spectral coding and reduce the coding complexity using a mesh partitioning technique. Our algorithm can be used for the progressive transmission and multi-representation applications of 3-D meshes.

However, we will have to research more to solve two problems in the future. First, we should try to solve the error propagation problem when the geometry information regenerate from dual graph information. Second, we should develop a good mesh partition scheme to reduce coding complexity.

6. ACKNOWLEDGEMENT

This work was supported in part by GIST, in part by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research-Center (RBRC) at GIST, and in part by the Ministry of Education (MOE) through the Brain Korea 21 (BK21) project.

References

[1] M. Deering, " Geometry compression," Proceedings of SIG- GRAPH , pp. 13-20, 1995.

[2] G. Taubin and J. Rossignac, "Geometric compression throu- gh topological surgery," ACM Transactions on Graphics vol. 17, pp. 84-115, 1998.

[3] C. Touma and C. Gotsman, " Triangle mesh compression," Proceedings of Graphics Interface, pp. 26-34, 1998.

[4] Z. Karni and C. Gotsman, " Spectral compression of mesh geometry," Proceedings of SIGGRAPH pp.279-286, 2000.

[5] Z. Karni and C. Gotsman, "3D mesh compression using fixed spectral basis," Proceedings of Graphics Interface, pp. 1-8, 2001.

[6] Z. Yan, S. Kumar, J. Li and C-C.J. Kuo, " Robust coding of 3D graphics models using mesh segmentation and data parti ioning," Proceedings of IEEE International Conference on Image Processing, pp. 25-28, 1999.

[7] S.Y. Kim, J.H. Ahn and Y.S. Ho, "View-dependent Transmission of Three-dimensional Mesh Models Using Hierarchical Partitioning," Proceedings of VICP, pp. 1928-1938, 2003.