Progressive Wavelets Compression for 3D models

Chong Zhao
Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
Email: czhao@cse.cuhk.edu.hk

Hanqiu Sun
Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong, China
Email: hanqiu@cse.cuhk.edu.hk

Kaihuai Qin
Computer Science and Technology
Tsinghua University
Beijing 100084, China
Email: qkh-dcs@tsinghua.edu.cn

***Abstract*—Progressive geometry compression is an efficient way to balance the performance of processing highly detailed 3D models and the limited resource of computation and network transmission. In this paper, we propose a novel progressive compression methodology based on the lifting wavelets for arbitrary highly detailed triangular meshes. By applying the global lifting subdivision wavelets that can decompose the meshes in linear time, our approach has the advantages of efficient computation and less memory usage. In contrast to the existing wavelets based compression approaches, most require much time costs to solve the high dimension system during the compression. The initial experiments showed the high efficiency and good compression quality of our proposed approach, useful for the mobile and online 3D graphics/gaming applications.**

***Index Terms*—Loop subdivision, subdivision wavelets, continuous inner product, global lifting, progressive compression**

I. Introduction

During the development of graphics technologies, the highly detailed models are widely used in the 3D applications on the internet and mobile devices. The models with millions vertices and faces not only consume the huge bandwidth of network transmission, but also challenge the computing power of clients, especially to the mobile devices. For these applications, the progressive compression is an efficient way to balance the good performance and limited resource. In this paper, we propose a progressive compression approach base on the lifting subdivision wavelet transform on triangular meshes. To gain better compression qualities, we exploit the global lifting and continuous inner product to construct the wavelets. The lifting wavelets transform we adopted can be converted to several steps, which are directly applied on vertices, thus our approach is more efficient with less memory cost.

In the following, Section 2 outlines the related work in subdivision wavelets and wavelets based geometry compression. Section 3 presents the wavelets transform we propose for geometry compression, and the initial experimental results using our approach. Finally, the summary of our work is given in Section 4.

II. Related Work

Since the classical wavelets are not suitable to be applied in the 3D domain, the most wavelets used in progressive geometry compression are constructed from the subdivision. The theory framework of subdivision wavelets was developed by Lounsbery et al [12]. They also presented several subdivision wavelets, which are generally used on the surface of arbitrary topology. The lifting scheme proposed by Swelden [19] is an important tool to construct subdivision wavelets, because it can generate new biorthogonal wavelets from the classic wavelets and lazy wavelets. Using local lifting and the discrete inner product, Bertram [1] constructed a biorthogonal wavelet based on the Loop subdivision. Li et al. [10] proposed an unlifted Loop subdivision wavelet by optimizing free parameters on the extended subdivisions. Wang et al. [22], [23] developed several lifting wavelets based on Catmull-Clark subdivision and $\sqrt{3}$ subdivision.

Based on subdivision wavelets, methods of geometry compression have been developed. Most of them are based on interpolating subdivision schemes [4], [7], [18], [6], such as Butterfly subdivision. By applying the loop subdivision based wavelets, Khodakovsky et al [9] proposed a progressive compression approach (short as PGC in the following) and develop a pure geometry coder, where the input model is remeshed and provides the best rate-distortion trade offs so far, when the user does not need to keep the original connectivity of the 3D mesh. They also provided a framework for progressive compression of arbitrary topology, which includes semi-regular wavelet transforms, vector valued wavelet coefficients, zerotree coding, entropy coding and subdivision based reconstruction. His scheme needs to solve a linear system that makes the forward transform slower than the inverse transform. Vallete et al. [21] presented compression scheme for triangular meshes, based on a wavelet multiresolution theory for irregular meshes. The method also needs to solve the linear system. Bertrem et al. [3] proposed a local lifting biorthogonal wavelets for the geometry compression of quadrilateral meshes, and their wavelet scheme is efficient and less memory cost. By exploiting integer arithmetic coding, the wavelet scheme can be used to lossless compression.

Different from the work based on subdivision wavelets, the geometry image proposed by Gu et al. [8] provided a way to apply the methodologies of image compression on surface. Peyre et al. [14] developed a geometry compression method based on the second generation bandlet bases. They decompose the surfaces in a bandlet basis after removing the geometry redundancy of orthogonal wavelet coefficients. Their method can be applied in geometry image and normal map compression.

III. Wavelets Transform

A. Wavelets on loop subdivision

In this paper, the loop subdivision is used as the fundamental subdivision to construct our wavelets transform for compression. Loop subdivision [11] is an efficient and widely used subdivision scheme on triangular meshes. And some popular geometry compression based on loop subdivision wavelets have been constructed. But most of them need to solve a linear system whose dimension depends on the number of control points of mesh. It make the computation very expensive. In order to improve the efficiency of compression, we construct the lazy wavelet transform from the subdivision first and increase the orthogonality of it through lifting in the following. The advantage of this approach is that the computation of wavelet transform is in-place and simplified by avoiding the huge linear system solving.

The refinement stencils of loop subdivision, showed in Figure 1, can be written as:

$$e' = \tfrac{3}{8}(v_0 + v_1) + \tfrac{1}{8}(v_2 + v_3)$$
$$v' = \alpha(n)v + \tfrac{1-\alpha(n)}{n} \sum_{i=1}^{n} v_i{}'$$

where $\alpha(n) = \frac{3}{8} + (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n})^2$, n is the valence of v. e is the edge point of loop subdivision at resolution j and v is the point at resolution j.

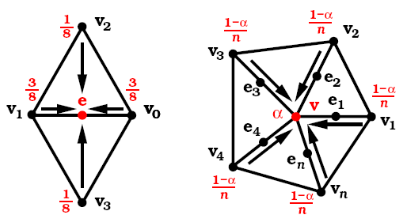

Fig. 1. The refinement stencils of loop subdivision.

Following the work [1], we can get the lazy wavelet transform stencils by transforming the loop subdivision stencils to:

$$e' = e + \tfrac{3}{8}(v_0 + v_1) + \tfrac{1}{8}(v_2 + v_3)$$
$$v' = \beta(n)v + \tfrac{1-\beta(n)}{n} \sum_{i=1}^{n} e_i{}'$$

where $\beta(n) = \frac{8}{5}\alpha(n) - \frac{3}{5}$, $\alpha(n) = \frac{3}{8} + (\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n})^2$, n is the valence of v. If e is initialized by zero, these stencils are equal to the loop subdivision. These stencils define a lazy wavelet synthesis. The basis function corresponding to e is lazy wavelet ψ and the basis function corresponding to v is scaling function ϕ. We can get wavelet analysis transform by inverting the stencils from the second one.

As the scaling functions and lazy wavelets are not sure to be orthogonal, the fitting qualities of lazy wavelet transform are poor. So we construct a new wavelet ψ' by lifting the corresponding lazy wavelet ψ locate at e with the scaling functions ϕ around e. To improve the orthogonality of wavelet transform, the new wavelet ψ' should be orthogonalized with the scaling functions around it.

$$\psi' = \psi + \sum_{i=0}^{n} w_i \phi_i \quad and \quad \langle \psi', \phi_i \rangle = 0. \tag{1}$$

B. Lifting scheme

To solve the system (1) and get the weights w_i, we should determine the inner product of lazy wavelet. For subdivision based lifting wavelets, discrete inner product is often used to simplify the computation. The idea of discrete inner product used in subdivision wavelets is based on the assumption that the scaling functions of finer resolution form an orthogonal basis without considering all correlation of finer-level coefficients. With this assumption, the mutual inner product of wavelets and scaling functions is defined as the sum of multiplications of corresponding coefficients (geometry coordinates of points) at finer resolution, and calculated directly from the subdivision stencils. It is really simple, but not accurate because the assumption may not be valid. Here, we employ bezier net to calculate the continuous inner product of lazy wavelet. The continuous inner product of lazy wavelet is defined as:

$$\langle \psi, \phi \rangle = \sum_{\delta \in \triangle(M)} \frac{1}{area(\delta)} \int_{x \in \delta} \psi(x)\phi(x)dx$$

where $\triangle(M)$ is the set of triangular faces of mesh M. Without considering the extraordinary points, the vertices of mesh can be mapped to the regular $\triangle^1$ grid, just as the Figure 2 showed. The basis functions of loop subdivision is relative to the vertices of mesh at resolution j. Suppose the basis function of loop subdivision at resolution j is ϕ^j, n is the size of support of scaling function, and $\phi^j = \sum_{i=1}^{n} P_i \phi^{j+1}$, according to the definition of subdivision, from the definition of lazy wavelet we have:

$$f = \sum_{k \in M^{j+1}} v^{j+1}\phi^{j+1} = \sum_{k \in M^j} v_k^j \phi^j$$
$$+ \sum_{k \in M^{j+1}} e_k \phi^{j+1} + \sum_{k \in M^{j+1}} \frac{1-\alpha(n)}{n} \sum_{i=1}^{n} e_i \phi^{j+1}$$

So, we can define the wavelet function of lazy wavelet as: $\psi^j = \phi^{j+1} + \frac{1-\alpha(n)}{n}\phi_k^{j+1}$, where $k = 0, 1$.

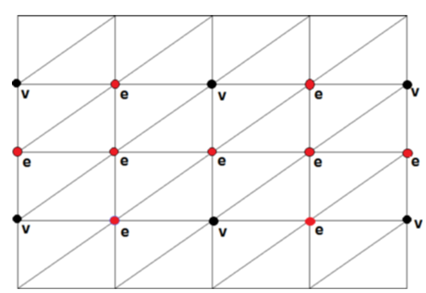

Fig. 2. Mapping the mesh to $\triangle^1$ grid.

The definition and calculation of basis function of loop subdivision, box spline B_{222}, can be found in [5]. Though

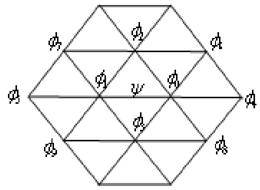

Fig. 3. Constructing a wavelet as linear combination of a lazy wavelet ψ and the scaling functions ϕ_i (i=0,...,9).

the computation of continuous inner product is more complex than the discrete inner product, the inner product can be precomputed as the computation is invariant of the geometry mesh. So it won't reduce the efficiency of wavelet transform. For high compression ratio, we adopt global lifting based on the continuous inner product. The mask of lifting is showed in Figure 3. The inner product of lazy wavelets are:

$$\begin{aligned}
&\langle \psi, \phi_0 \rangle = \langle \psi, \phi_1 \rangle = 0.5489, \\
&\langle \psi, \phi_2 \rangle = \langle \psi, \phi_3 \rangle = 0.1627, \\
&\langle \psi, \phi_6 \rangle = \langle \psi, \phi_7 \rangle = \langle \psi, \phi_6 \rangle = \langle \psi, \phi_7 \rangle = 0.1034, \\
&\langle \psi, \phi_4 \rangle = \langle \psi, \phi_5 \rangle = 0.0918, \\
&\langle \phi_0, \phi_0 \rangle = 2.0942, \quad \langle \phi_0, \phi_1 \rangle = 0.8306, \\
&\langle \phi_0, \phi_5 \rangle = 0.0406, \quad \langle \phi_0, \phi_7 \rangle = 0.0406, \\
&\ldots
\end{aligned}$$

With the inner product of lazy wavelet,we can solve the linear system (1) and get the weights for lifting. The new rules of the wavelet synthesis can be constructed from the lazy wavelets:

$$\begin{aligned}
v_i &= v_i + w_i e \quad for\ \forall e, \forall i = 0, ..., 9; \\
e' &= e + \frac{3}{8}(v_0 + v_1) + \frac{1}{8}(v_2 + v_3) \\
v' &= \beta(n)v + \frac{1-\beta(n)}{n}\sum_{i=1}^{n} e_i{}'
\end{aligned}$$

The wavelet analysis rules can be got by reversing the wavelet synthesis rules from the last mask.

C. Quantization

For high compression ratio, the wavelet coefficients should be quantized (round to integers), which will introduce quantization errors. Since it is difficult to control the effect of quantization errors after the biorthogonal wavelets transform, we perform the quantization by computing wavelets in integer arithmetic [13]. We assume that the coordinates of control points at the highest resolution have finite precision and can be scaled up to integer, We also scale the lifting weights to integers. Then we round the wavelets in each transform to integers. The coefficients are represented by integer numbers.

Fig. 4. Comparison of compressed meshes, from left to right: 1/16 bit/vertex, 1/8 bit/vertex, 1/4 bit/vertex, 1/2 bit/vertex, 1 bit/vertex and the original mesh.

The integer wavelets synthesis are defined as:

$$\begin{aligned}
v_i &= v_i + w_i e \quad for\ \forall e, \forall i = 0, ..., 9 \\
e' &= e + [\frac{3}{8}(v_0 + v_1) + \frac{1}{8}(v_2 + v_3)] \\
v' &= \beta(n)v + \frac{1-\beta(n)}{n}\sum_{i=1}^{n} e_i{}'
\end{aligned}$$

where the rounding operator $[.]$ means returning the integer closest to its argument. The wavelets analysis can be got by reversing the synthesis.

$$\begin{aligned}
v &= \frac{1}{\beta(n)}(v' - \frac{1-\beta(n)}{n}\sum_{i=1}^{n} e_i{}' \\
e &= e - [\frac{3}{8}(v_0 + v_1) - \frac{1}{8}(v_2 + v_3)] \\
v_i &= v_i - w_i e \quad for\ \forall e, \forall i = 0, ..., 9
\end{aligned}$$

D. Experimental results

The precomputed lifting weights can greatly increase the efficiency of compression. Table I shows precomputed weights when the valence of vertex is 6.

TABLE I
THE PRECOMPUTED WEIGHTS OF LIFTING OPERATIONS, WHEN THE VALENCE N=6.

w_0	w_1	w_2	w_3	w_4
-0.2805	-0.2805	0.1353	0.1353	0.0612
w_5	w_6	w_7	w_8	w_9
0.0612	0.0035	0.0035	0.0035	0.0035

We compare our approach with PGC [9], which is widely used in geometry compression by applying the similar geometry coder. Table II shows the PSNR (peak signal to noise ratio) at different bitrates (measured by bits/vertex), where $\mathrm{PSNR} = 20 \times \log_{10} peak/d$, d is the L_2 error and peak is the bounding box diagonal. From the testing results, we see

TABLE II
COMPRESSION: (BIT/VERTEX) AND CORRESPONDING PSNR.

horse		1/4	1/2	1	3
	Ours	61.75	66.58	72.27	83.74
	PGC	65.73	69.85	74.89	83.37
horse		6	9	12	15
	Ours	90.75	97.34	102.70	106.87
	PGC	89.64	95.13	99.59	103.02
venus		1/4	1/2	1	3
	Ours	61.25	66.64	71.18	79.55
	PGC	64.15	68.43	72.38	80.16
venus		6	9	12	15
	Ours	84.97	88.41	92.36	97.02
	PGC	85.33	89.27	94.07	99.27

that our approach is similar in compression ratio to PGC in most cases, but much better in time performance. Only at low bitrates, it is about 3-4 dB inferior to PGC's. Figure 4 shows the meshes restored from compressed data at different bitrates.

We tested the compression efficiency of our approach and PGC by using a PC equipped with Intel Core 2 mobile CPU at 1.83 HZ and 1.5 G memory. The time costs of compression without coding are listed in the Table III. Because the lifting operation is executed on each vertex and the time complexity of each operation is O(1), the time complexity of lifting operations of wavelet transform only depends on the number of vertices. The efficiency of our approach is much better than PGC. The experimental results have proven it.

TABLE III
THE TIME COST OF COMPRESSION IN SECONDS (WITHOUT ZEROTREE CODING).

	Horse (112642 pt)	Venus (198658 pt)	Feline (258046 pt)
PGC	2.927 sec.	4.719 sec.	7.203 sec.
Ours	0.188 sec.	0.312 sec.	0.438 sec.

IV. SUMMARY

In this paper, we propose a novel approach in efficient progressive geometry compression for arbitrary 3D models. We adopt the lifting wavelets transform based on loop subdivision, which makes our approach highly efficient with high compression ratio and less memory cost. The efficient features of our approach are especially suitable for the mobile and/or on-line 3D applications with limited resource. Further, we will study on how to mostly optimize the compression algorithms and their comparative performance.

ACKNOWLEDGEMENT

This work was supported by RGC research grant (ref. 415806), UGC direct grant for research (no.2050423), the National Natural Science Foundation of China (no. 60773144, no.60973101) and the Beijing Natural Sciences Foundation (no.4072015). We gratefully thank the Caltech Multi-Res Modeling Group for publishing their PGC executables and data files on the web for comparison purposes.

REFERENCES

[1] Bertram, M.: Biorthogonal loop-subdivision wavelets. Computing **72**(1-2), 29–39 (2004)
[2] Bertram, M., Duchaineau, M., Hamann, B., Joy, K.: Bicubic subdivision-surface wavelets for large-scale isosurface representation and visualization. In: VIS '00: Proceedings of the conference on Visualization '00, pp. 389–396 (2000)
[3] Bertram, M., Duchaineau, M., Hamann, B., Joy, K.: Generalized b-spline subdivision-surface wavelets for geometry compression. IEEE Transactions on Visualization and Computer Graphics **10**(3), 326–338 (2004)
[4] Bonneau, G.P.: Optimal triangular haar bases for spherical data. In: VIS '99: Proceedings of the conference on Visualization '99, pp. 279–284 (1999)
[5] Chui, C.: Multivariate splines (1988)
[6] Eck, M., Derose, T., Duchamp T.and Hoppe, H., Lounsbery, M., Stuetzle, W.: Multiresolution analysis of arbitrary. Computer Graphics **29**, 173–182 (1995)
[7] Gross, M.H., Staadt, O.G., Gatti, R.: Efficient triangular surface approximations using wavelets and quadtree data structures. IEEE Transactions on Visualization and Computer Graphics **2**(2), 130–143 (1996)
[8] Gu, X., Gortler, S., Hoppe, H.: Geometry images. In: Proc. of SIGGRAPH, pp. 355–361 (2002)
[9] Khodakovsky, A., Schröder, P., Swelddens, W.: Progressive geometry compression. In: SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 271–278 (2000)
[10] Li, D., Qin, K., Sun, H.: Unlifted loop subdivision wavelets. In: PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, pp. 25–33. IEEE Computer Society (2004)
[11] Loop, C.T.: Smooth subdivision surfaces based on triangles (1987)
[12] Lounsbery, M.: Multiresolution analysis for surfaces of arbitrary topological type. PhD. thesis. (1994)
[13] Moffat, A., Neal, R.: Arithmetic coding revisited. In: DCC '95: Proceedings of the Conference on Data Compression, p. 202. IEEE Computer Society, Washington, DC, USA (1995)
[14] Peyré, G., Mallat, S.: Surface compression with geometric bandelets. In: SIGGRAPH '05: ACM SIGGRAPH 2005 Papers, pp. 601–608 (2005)
[15] Samavati, F., andR.H. Bartels, N.M.A.: Multiresolution surfaces having arbitrary topologies by a reverse doo subdivision method. Computer Graphics Forum (2), 121–136 (2002)
[16] Samavati, F., Bartels, R.: Multiresolution curve and surface editing: reversing subdivision rules by least-squares data fitting. Computer Graphics Forum (2), 97–119 (1999)
[17] Schröder, P., Sweldens, W.: Spherical wavelets: efficiently representing functions on the sphere. In: SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques, pp. 161–172 (1995)
[18] Staadt, O., Gross, M., Weber, R.: Multiresolution compression and reconstruction. Visualization Conference, IEEE **0**, 337 (1997)
[19] Sweldens, W.: The lifting scheme: A custom-design construction of biorthogonal wavelets. Appl. Comput. Harmon. Anal. **3**(2), 186–200 (1996)
[20] Valette, S., Prost, R.: Wavelet-based multiresolution analysis of irregular surface meshes. IEEE Transactions on Visualization and Computer Graphics **10**(2), 113–122 (2004)
[21] Valette, S., Prost, R.: Wavelet-based progressive compression scheme for triangle meshes: Wavemesh. IEEE Transactions on Visualization and Computer Graphics **10**(2), 123–129 (2004)
[22] Wang, H., Qin, K., Sun, H.: $\sqrt{3}$-subdivision-based biorthogonal wavelets. IEEE Transaction on Visualization and Computer Graphics **13**(5), 914–925 (2007)
[23] Wang, H., Qin, K., Tang, K.: Efficient wavelet construction with catmull-clark subdivision. Vis. Comput. **22**(9), 874–884 (2006)