Wyner-Ziv Coding of 3D Dynamic Meshes

Chao Chen[a] , Qifei Wang[a], Qionghai Dai[a], Zixiang Xiong[b], Xiaodong Liu[a]

[a]Broadband Networks & Digital Media Laboratory, Dept. of Automation, Tsinghua Univ.;
[b]Dept.ECE, Texas A&M University, College station, TX77843, U.S.

ABSTRACT

Existing 3-D dynamic mesh compression methods directly explore temporal redundancy by predictive coding and the coded bitstreams are sensitive to transmission errors. In this paper, an efficient and error-resilient compression paradigm based on Wyner-Ziv coding (WZC) is proposed. We first apply an anisotropic wavelet transform (AWT) on each frame to explore their spatial redundancy. Then the wavelet coefficients of every frame are compressed by a Wyner-Ziv codec which is composed of a nested scalar quantizer and a turbo codes based Slepian-Wolf codec. Benefiting from the inherent robustness of WZC, the proposed coding scheme can alleviates the problem of error-propagation associated with conventional predictive coding scheme. Furthermore, based on wavelet transform, our method can be extended to support progressive coding which is desirable for the streaming of 3D meshes. Experimental results show that our scheme is competitive with other compression methods in compression performance. Moreover, our method is more robust when transmission error occurs.

Keywords: 3-D dynamic mesh, Nested scalar quantization, Wyner-Ziv coding, Slepian-Wolf coding

1. INTRODUCTION

3-D meshes are widely used in computer graphics related applications. Generally, a static 3-D mesh consists of thousands of vertices which require a huge amount of data to describe their geometrical positions and their relations. So far, many static 3-D mesh compression algorithms have been proposed.[1] Besides, dynamic 3-D mesh which consists of a series of static meshes is used to model the movement of 3-D objects. It requires a much larger amount of data than the static one.

Lengyel first proposed his segmentation-based coding scheme in 2. This scheme approximates the motion of the segmented mesh frame by an affine transform and only encodes the approximation residuals and transform parameters. In 3 and 4, vertex-wise motion vector prediction was used to explore spatial-temporal redundancy. In 5, Zhang et al employed octree to cluster the motion vectors and reduce the spatial redundancy. K.Muller et al. incorporated Zhang's algorithm in their design and extended MPEG4 3D static mesh coder(MPEG 3DMC)[6] to dynamic mesh compression.[7] In 8 and 9, Principle Component Analysis (PCA) was introduced. Such methods treat a mesh sequence as a large matrix and only encode the most important PCA coefficients. In 10, Briceno et al. proposed a scheme named geometry video. By mesh cutting and global parameterization, the original mesh sequence is converted to a 2-D video and thus can be compressed with 2-D video coding technologies. In 11, Payan et al. introduced wavelet transform to 3-D dynamic mesh coding. A temporal wavelet transform is employed to reduce the temporal redundancy of successive frames. However, the spatial redundancy was not taken into account. Guskov et al. proposed another wavelet-based scheme which supports progressive compression.[12] Different from 11, it applies a spatial anisotropic wavelet transform on each frame to explore the spatial redundancy via multi-resolution analysis. To reduce the temporal redundancy, wavelet coefficients of each frame are predictively encoded. In 13, Cho et al. proposed a similar wavelet-based progressive compression method which can further support lossless compression. Besides, Interpolator Compression algorithm of MPEG4 Animation Framework eXtention (AFX-IC)is the state of art standard approach.[14] In this paper, it is used as benchmark to evaluate the efficiency of our codec.

All the above methods only consider the isomorphic mesh sequences, where the number of vertices and their connectivity are invariant through all the frames. For non-isomorphic mesh sequence, Yang et al. proposed a progressive coding scheme based on semi-regular mesh representation and spatiotemporal wavelet transform.[15] In our proposed scheme, we still focus on isomorphic dynamic meshes.

Visual Communications and Image Processing 2008, edited by William A. Pearlman, John W. Woods, Ligang Lu,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 6822, 68221V, © 2008 SPIE-IS&T · 0277-786X/08/$18

There is a common shortcoming in all existing schemes. They are all fragile in the context of transmission errors. For PCA-based schemes, if an important PCA coefficient is lost during transmission all frames will be erroneously reconstructed. For the other compression schemes which exploit temporal redundancy by predictive encoding at the encoder, the successful decoding of current frame is dependent of the correct decoding of all preceding frames. Besides, all those algorithms utilize entropy coding such as arithmetic coding or variable length coding to compress output symbols of the quantizer. Hence, their output bitstreams are very sensitive to bit errors. Even sparse transmission errors will lead to error propagation and severe degradation of the decoded 3-D mesh's visual quality.

In this paper, a robust and efficient coding scheme is proposed. The key idea of the proposed scheme is to compress the wavelet coefficients of each frame based on distributed source coding(DSC) principles.

DSC refers to the compression of two or more dependent sources which cannot access each other. Each source is separately encoded and jointly decoded. Slepian and Wolf theoretically proved that, for lossless coding, the compression performance of DSC can be as good as conventional joint coding scheme.[16] Wyner and Ziv further extended this theory to lossy source coding with decoder side information(WZC), a special case of DSC. In 17, they showed that when the source and side information are two zero-mean and stationary Gaussian memoryless sources, under MSE distortion metric, WZC has the same rate-distortion performance as conventional coding scheme. Spurred by the invention of near capacity channel codes, several practical code designs of DSC have been proposed.[18–20] In recent years, the application of DSC attracts more and more attention especially in sensor networks[19, 20] and video coding.[21, 22]

In general, there are two orientations about putting DSC into application. One is to focus on constructing low encoding complexity codec in power-constraint wireless sensor networks and uplink video communication systems. However, due to the difficulty of correlation estimation at decoder, current complexity-constraint codec design can not meet the need of practical applications. The other orientation is utilizing WZC to enhance the robustness of bitstreams. A.Sehgal et al. implemented a robust Wyner-Ziv video coding framework by periodically transmitting coset information to prevent error-propagation of standard predictive coding. B.Girod et al. used the output parity bits of a Wyner-Ziv codec to protect standard video bitstream from transmission errors.[21] In 23, Xu and Xiong employed layered WZC[24] to alleviate the error drifting problem in standard scalable video codec. R.Puri et al.'s PRISM codec also exhibit its robustness against transmission errors.[22] In this paper, we employ layered WZC[24] to enhance the bitstream's robustness.

Specifically, the proposed codec consists of anisotropic wavelet transform(AWT), nested scalar quantization(NSQ)[25] and turbo codes based Slepian-Wolf coding(SWC). Anisotropic wavelet transform is used to explore the spatial redundancy within each frame. The combination of NSQ and turbo codes based SWC forms the Wyner-Ziv part of our codec, which explores the temporal redundancy of successive frames. NSQ is essentially a 1-D lattice binning scheme which assigns a bin index to each input wavelet coefficient. Then the bin indices are extracted in bit planes and further compressed with SWC. The robustness of our codec partly results from NSQ which independently generates the bin indices of current mesh frame rather than quantization indices of motion vectors. Besides, SWC's probabilistic decoding algorithm provides further robustness over deterministic decoding of conventional entropy coding scheme.

Another compelling attribute of the proposed scheme is that it supports progressive coding. Progressive coding allows the transmission and rendering of 3-D dynamic mesh on different level of details (LOD). However, most existing algorithms do not support progressive coding.[2–5, 7–10, 14] With anisotropic wavelet transform, the proposed scheme conducts multi-resolution analysis on each frame and can be extended to support progressive mesh coding.

The rest of the paper is organized as follows. In Section 2, the proposed compression scheme is detailed. Then the experimental results are presented in Section 4, followed by the conclusion in Section 5.

2. PROPOSED SCHEME

The block diagram of proposed codec is shown in Fig.1. It consists of 4 main components, i.e. AWT, NSQ, SWC and Correlation Estimation. In the following, we first detail these main components and then describe how the codec works.

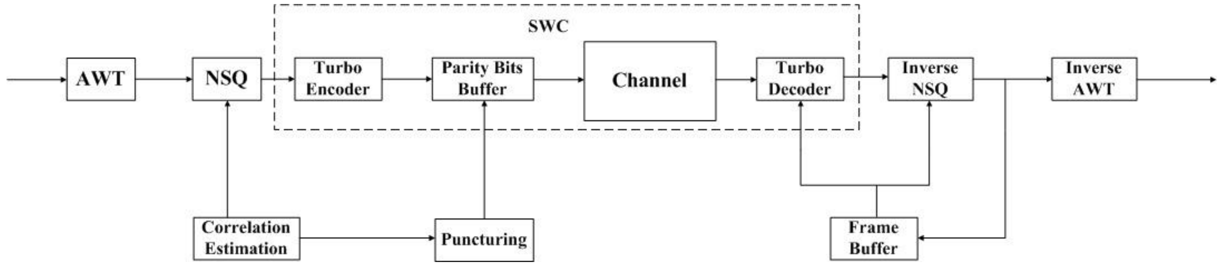


Figure 1. block diagram of the proposed codec

2.1 AWT

To explore spatial redundancy and support progressive coding, wavelet transform is employed to perform multi-resolution analysis (MRA) on every frame. The first wavelet-based MRA scheme was proposed by Lounsbery.[26] However, it can only deal with regular meshes in which the degrees of all inner vertices are the same. As for the more general case of irregular meshes, Daubechies and Guskov introduced a wavelet transform based on non-uniform subdivision.[27,28] In 12, Guskov further modified it to an anisotropic wavelet transform which is suitable for compression. We also adopt this anisotropic wavelet transform.

The main idea of AWT is as follows: it utilizes Hoppe's Progressive Mesh scheme[29] to construct a hierarchy of progressively simplified mesh. Meanwhile, as shown in Figure 2(a), AWT computes a wavelet coefficient every time a vertex is removed. The wavelet coefficient corresponding to a removed vertex is the difference between its original position and the predicted position from the unremoved vertices by an anisotropic predictor. Because the geometry positions of vertices are three-dimensional data, wavelet coefficients are also three dimensional.* As shown in the histogram of wavelet coefficients in Figure 2(b), AWT is powerful in exploring the spatial redundancy.

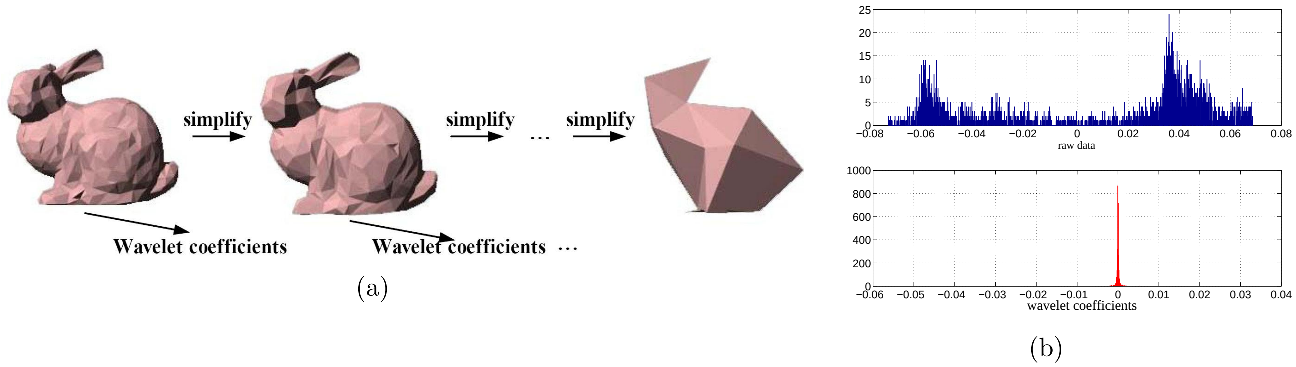


Figure 2. (a)wavelet transform and progressive mesh hierarchy.(b)histogram of raw data and wavelet coefficients (X component of the 15th frame in **Cow** sequence,)

In the proposed codec, the first frame is used as training set to generate the parameters of anisotropic predictors in AWT. Then, these predictor parameters are used to get wavelet coefficients of subsequent frames. Specifically, the first frame is compressed with MPEG 3DMC[6] and sent to the decoder. It is decoded in both encoder and decoder to make sure that both sides have same training set. Then, a progressive mesh hierarchy is constructed on the decoded frame to compute anisotropic predictor parameters. Hence, both encoder and decoder hold the same parameters.

For more details about how to compute anisotropic predictor parameters from progressive mesh, please refer to 12.

*Because the three components of the coefficient are independently coded, we term coefficient itself and any of its components all as wavelet coefficients.

2.2 NSQ

With AWT, the raw data of mesh frames are transformed into wavelet coefficients. Then NSQ is used to quantize these coefficients in a Wyner-Ziv manner.

In our scenario, the wavelet coefficients of current frame $X = \{x_1, ...x_n\}$ are coded with previously reconstructed frame $Y = \{y_1, ...y_n\}$ as side information. Then, NSQ partitions the codebook of a fine uniform scalar quantizer into several uniformly nested coset codes. For sample x_i, it is quantized by the fine codebook but only the index of the coset to which x_i belongs is sent to the reconstructor. NSQ reconstructs x_i as the closest codeword to y_i in the indexed coset. Note that the side information y_i is only accessed in reconstruction. In this sense, NSQ can be viewed as a prototype of WZC.

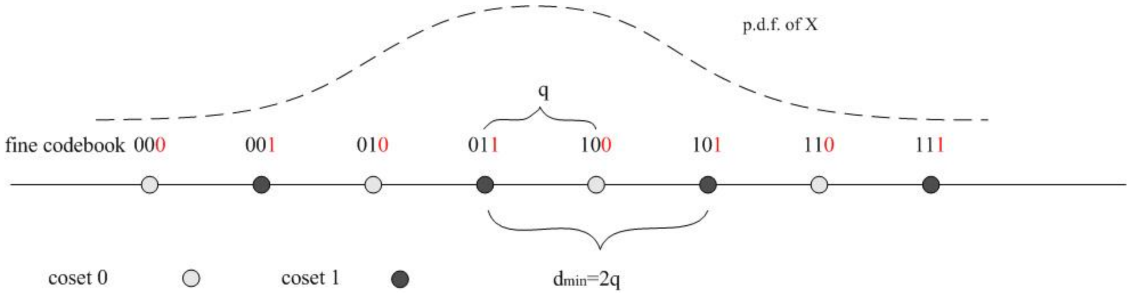


Figure 3. NSQ with nesting ratio of 2

Denote q and d_{min} as the minimum distance of the fine codes and coset codes, respectively. The number of cosets is termed as nesting ratio N. We have $d_{min} = N \times q$. An example of NSQ with nesting ratio of 2 is shown in Fig. 3. In this example, the least significant bit can be directly used as coset index.

For NSQ, it can be proved that if

$$|x_i - y_i| < d_{min}/2, \tag{1}$$

x_i can be reconstructed correctly no matter y_i is corrupted or not.[25] This important property contributes to the robustness of our codec.

2.3 SWC

For NSQ with nesting ratio N, the coset index can be represented with a binary number of $\Gamma = \lceil log_2 N \rceil$ bits. These coset indices are then split into bit planes. There is still redundancy between these bit planes and side information Y which can be further exploited by SWC. Here, the role SWC plays is similar to entropy coding in predictive coding scheme.

Denote the bit planes by $B_\Gamma(X), ...B_1(X)$. They are layered coded from the most significant one to the least significant one. For kth bit plane, the theoretical bit rate of SWC is

$$H_k = H(B_k(X)|Y, B_{k-1}(X), ..., B_1(X)), \tag{2}$$

in which $B_{k-1}(X), ..., B_1(X)$ are previously decoded bit planes. They are utilized together with Y for SWC. Because $H(B_\Gamma(X), ...B_1(X)|Y) = H(B_1(X)|Y) + H(B_2(X)|Y, B_1(X)) + ... + H(B_\Gamma(X)|Y, B_{\Gamma-1}(X), ..., B_1(X)) = \sum_1^\Gamma H_k$, no rate loss is introduced theoretically by layered coding.

So far, most proposed practical SWC schemes are based on near-capacity channel codes such as turbo or LDPC codes.[30–32] Different from the deterministic decoding scheme in conventional entropy coding, the decoding scheme of channel codes is probabilistic. Hence, SWC could provide further robustness against bit errors.

In our scenario, the number of bits in a bit plane is about 10^3 to 10^4 [†] and we employ the turbo based scheme in our design. Specifically, we encode current bit plane $B_k(X)$ with a turbo encoder. The turbo code

[†] In AWT, each vertex corresponds to a wavelet coefficient. Generally, a mesh frame contains 10^3 to 10^4 vertices and thus produces the same number of wavelet coefficients

consists of two constituent [1,17/13] convolutional codes. Each input bit results in two parity bits. These parity bits are then periodically punctured and sent to the decoder. The decoder uses previously decoded bit planes $B_{k-1}(X), ..., B_1(X)$ and side information Y to generate $\hat{B}_k(X)$ as the prediction of $B_k(X)$. Then the received parity bits are used to "correct" $\hat{B}_k(X)$ back to $B_k(X)$. Here, we assume BSC channel for BCJR decoding algorithm[33] and employ puncture scheme to control the number of transmitted parity bits and adjust the compression ratio of SWC. The extent to which the parity bits are punctured and the prediction of $B_k(X)$ depend on the correlation estimation scheme detailed in the next subsection.

2.4 Correlation Estimation

As in most existing Wyner-Ziv codec, correlation estimation is a crucial part in our design. In the scenario of video coding, motion estimation is widely used. However, it requires very high computation complexity which renders it unsuitable for complexity-constraint Wyner-Ziv codec design. In our design, since there is no motion estimation need to be involved[‡], we can directly estimate the correlation at encoder without introducing too much complexity.

To accurately model the correlation, we not only generate side information Y in the decoder but also in the encoder. Every frame is decoded at the encoder and stored in a frame buffer. Hence, the encoder can access both current frame X and the side information Y for correlation estimation.

The correlation between X and Y is modeled with $Y = X + Z$, where X and Z are independent zero-mean Gaussian sources. Denote the variances of X, Y and Z by σ_X^2, σ_Y^2 and σ_Z^2 respectively. We have $\sigma_Y^2 = \sigma_X^2 + \sigma_Z^2$, in which σ_Z^2 can be estimated by $\sigma_Z^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2$. Hence, given y_i, the conditional distribution of x_i is

$$p(x_i|y_i) = \frac{1}{\sqrt{2\pi}\sigma_z} exp\{-\frac{(x_i - y_i)^2}{2\sigma_z^2}\}. \tag{3}$$

With the assistance of this correlation model, x_i can be numerically estimated from y_i. In our codec, correlation estimation is employed in following aspects:

Quantization In our implementation of NSQ, we fix the step-size q and select the nesting ratio N such that equation (1) is satisfied i.e.

$$N > 2\max_i |x_i - y_i|/q, \tag{4}$$

For reconstruction, we numerically estimate x_i as the centroid of $p(x_i|y_i)$ over the region indexed by $B_\Gamma(x_i)$, $B_{\Gamma-1}(x_i), ..., B_1(x_i)$, i.e.

$$\hat{x_i} = \int_{B_\Gamma, B_{\Gamma-1}, ..., B_1} \frac{x}{\sqrt{2\pi}\sigma_z} exp\{-\frac{(x - y_i)^2}{2\sigma_z^2}\}dx, \tag{5}$$

Parity bits Allocation SWC adjusts the compression ratio via puncturing. In the one hand, the more bits are punctured, the higher compression ratio SWC achieves. In the other hand, severe puncture will result in incorrect decoding. Hence, the encoder should estimate how many parity bits should be sent for current bit plane $B_k(X)$ given Y and $B_{k-1}(X), ..., B_1(X)$. Specifically, we numerically compute $p(B_k(x_i) = 1|y_i, B_{k-1}(x_i), ..., B_1(x_i))$ and $p(B_k(x_i) = 0|y_i, B_{k-1}(x_i), ..., B_1(x_i))$ via integrating $p(x_i|y_i)$ over the regions indexed by $B_k(x_i), ..., B_1(x_i)$,i.e.

$$p(B_k(x_i) = 1|y_i, B_{k-1}(x_i), ..., B_1(x_i)) = \int_{B_k=1, B_{k-1}, ..., B_1} \frac{1}{\sqrt{2\pi}\sigma_z} exp\{-\frac{(x - y_i)^2}{2\sigma_z^2}\}dx, \tag{6}$$

and

$$p(B_k(x_i) = 0|y_i, B_{k-1}(x_i), ..., B_1(x_i)) = \int_{B_k=0, B_{k-1}, ..., B_1} \frac{1}{\sqrt{2\pi}\sigma_z} exp\{-\frac{(x - y_i)^2}{2\sigma_z^2}\}dx, \tag{7}$$

[‡] Different from images or video frames, mesh frames are irregularly sampled signals which can not be directly modeled by motion estimation scheme.

Then, the Log-Likelihood Ratio(LLR)s $LLR(B_k(x_i)) = \frac{log(p(B_k(x_i)=1|y_i,B_{k-1}(x_i),...,B_1(x_i)))}{log(p(B_k(x_i)=0|y_i,B_{k-1}(x_i),...,B_1(x_i)))}$ of every bits in current bit plane are obtained. The hard-decision of LLR serves as prediction of current bit plane $\hat{B}_k(x_i)$, i.e.

$$\hat{B}_k(x_i) = \begin{cases} 1 & if LLR(B_k(x_i)) \geq 0 \\ 0 & if LLR(B_k(x_i)) < 0 \end{cases} \tag{8}$$

In the decoder, $\hat{B}_k(x_i)$ is used for decoding while in the encoder, it is used for bit allocation. By computing $P_e = p(B_k(x_i) \neq \hat{B}_k(x_i))$, $H(B_k|Y, B_{k-1}, ..., B_1)$ is estimated by

$$H(B_k|Y, B_{k-1}, ..., B_1) = -P_e log_2 P_e - (1 - P_e) log_2 (1 - P_e). \tag{9}$$

In practice, SWC is not as efficient as (9) promises. Hence, we use (9) as a conservative estimation. To secure successful decoding, a **doping** scheme is employed. In particular, we initialize the compression ratio of SWC as $1/H(B_k|Y, B_{k-1}, ..., B_1)$. Then, current bit plane is tentatively decoded in the encoder. If there is bit error in decoding, we doping more parity bits by adjusting our puncture scheme until all bits are successfully decoded.

The flow chart of our encoding algorithm including correlation estimation, Slepian-Wolf encoding and doping is shown in Fig.4. Note that all the processes are conducted in the encoder.

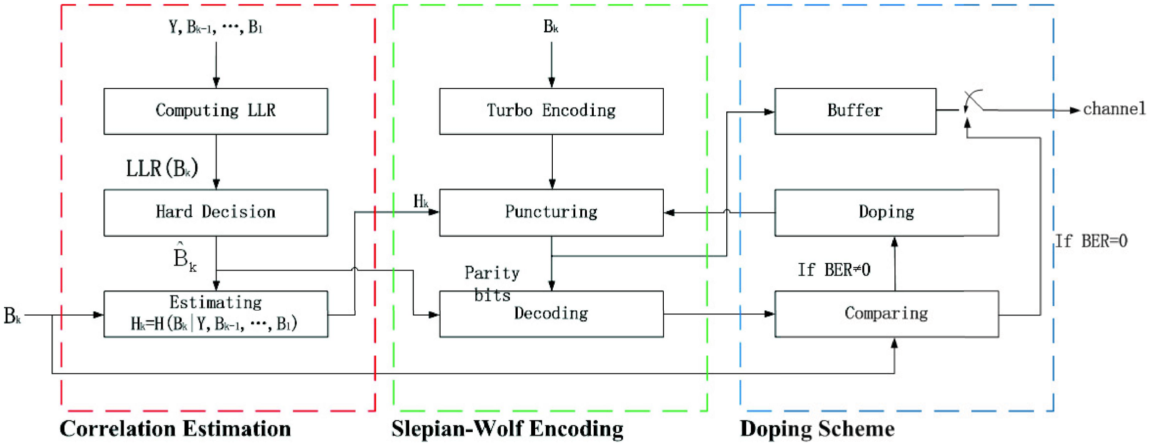


Figure 4. flow char of encoding algorithm

2.5 Overall Scheme

2.5.1 encoding

The encoding process includes two steps:

Step 1 The first frame is encoded and decoded with MPEG 3DMC codec. Then, a progressive mesh hierarchy is constructed on the decoded frame to compute anisotropic predictor parameters. Meanwhile, this compressed frame is sent to the decoder.

Step 2 In this step, the subsequent frames are encoded. For each frame, it is firstly transformed to wavelet coefficients with the predictor parameters generated in step1. Secondly, wavelet coefficients are quantized with NSQ and split into bit planes. These bit planes are then layered coded with SWC.

In the encoding process, the nesting ratio of NSQ and rate of turbo codes assigned to each bit plane is determined by the correlation estimation process. Once the coding of a bit plane is completed, it is stored for correlation estimation of the next bit plane.

2.5.2 decoding

Upon the first frame is received and decoded, it is used in the decoder to construct a progressive mesh hierarchy and computes predictor parameters, thus both encoder and decoder hold the same AWT parameters. Then, with the wavelet coefficients of the previously decoded frame as side information, the bit planes of current frame is decoded layer by layer. When all the bit planes are decoded, wavelet coefficients of current frame are reconstructed with equation (5). These coefficients are then inversely transformed by AWT as a decoded mesh frame.

3. EXPERIMENTAL RESULTS

3.1 Compression Performance

The compression performance of the proposed Wyner-Ziv 3D dynamic mesh compression(WDMC) scheme is tested on **Cow** sequence. This sequence is composed of 204 mesh frames. Each frame consists of 2904 vertices. In Fig.5, the rate-distortion performance of WDMC is compared with the algorithms reported in 12 and 9. The algorithm in 12 is a AWT-based predictive coding scheme thus is actually a non-distributed version of WDMC. For brevity, this algorithm is called AWC. The algorithm proposed in 9 is a typical PCA-based algorithm and is simply called KG. Besides, the state of art standard approach AFX-IC[14] is also employed as benchmark. For comparison, Karni and Gotsman's distortion metric (KG-error)[9] is used to measure the distortion. The step-size q of NSQ is set as 10^{-6}. The nesting ratio N is 2^{20}. Hence, for every frame, 20 bit planes need to be coded with layered SWC. Different operation points on the R-D curve are obtained via dropping different number of least significant bit planes. For the 6 operation points in Fig.5, we respectively drop 8,7,6,5,4 and 3 bit planes.

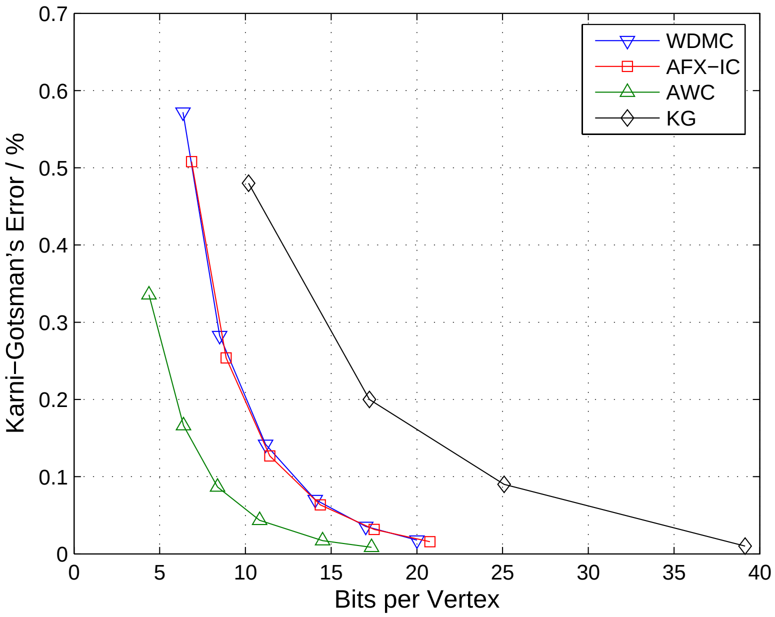


Figure 5. Rate-distortion curve for **Cow** sequence

As shown in Fig.5, under the same distortion, WDMC generally costs 3-4 bpv (bits per vertex) more than AWC does. The performance gap is mainly resulted from the inefficiency of our SWC implementation. Averagely, for every bit plane, our turbo codes based SWC costs 0.1-0.2 bits more than conventional entropy coding does. Meanwhile, it is noted that WDMC performs as good as AFX-IC and outperforms KG algorithm.

The experiment is conducted on **Dance** sequence as well. The settings for NSQ is the same as in Cow sequence. We respectively drop 6,5,4,3 and 2 bit planes to obtain the 5 operation points. This sequence contains

201 mesh frames and each frame consists of 7061 vertices. The performance is compared with AFX-IC in Fig.6. WDMC averagely saves 1 bpv over AFX-IC under the same distortion. It illustrates that WDMC is competitive with other non-distributed coding scheme even just in a viewpoint of compression.

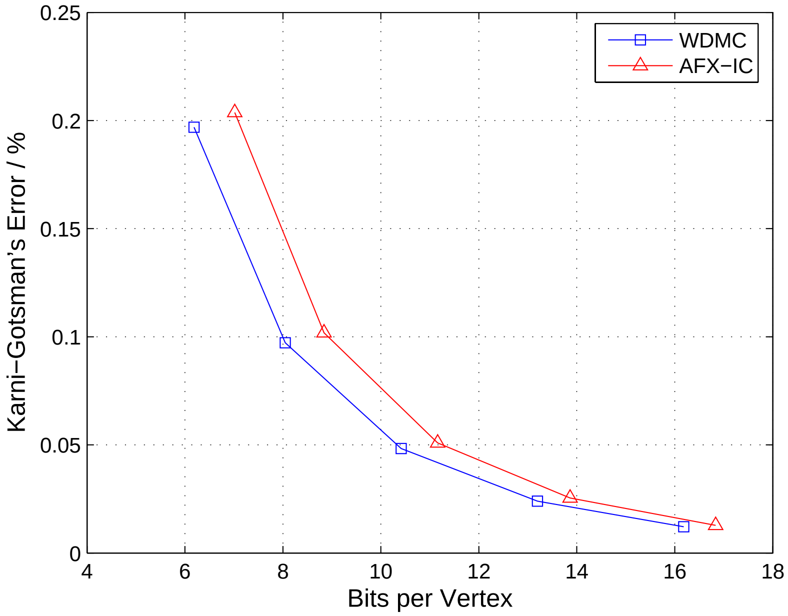


Figure 6. Rate-distortion curve for **Dance** sequence

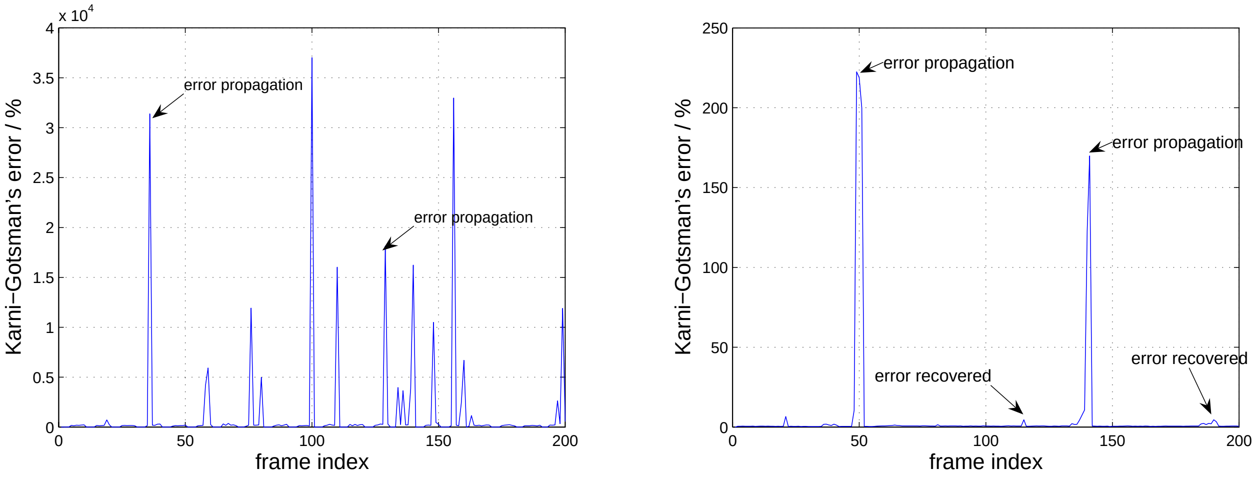
3.2 Robustness Simulations

To compare the robustness of the proposed codec with the state of art standard approach AFX-IC, bit errors are introduced to the bitstreams of both codecs. For fair comparison, the experiment is conducted on **Cow** sequence because, on this sequence, both schemes have nearly the same rate-distortion performance. To alleviate error propagation, the sequence is uniformly partitioned to Group of Meshes(GOM). We calculate the number of successfully decoded frames for both schemes under different bit error rate(BER). The experimental results is listed in Tab.1.

BER	successfully decoded frames / WDMC	successfully decoded frames / AFX-IC
10^{-4}	628	131
8×10^{-5}	704	130
5×10^{-5}	766	135
2×10^{-5}	838	342
10^{-5}	941	549

Table 1. simulation on robustness of WDMC

In the experiment, the length of GOM is 10. The key value quantization parameter of AFX-IC is set as 8. For WDMC, 8 least significant bit planes are dropped. Hence, both schemes have similar rate-distortion performance. For every BER, 1000 frames are used for simulation. The number of successfully decoded frames is calculated by counting the number of frames whose KG-error is less than 5%. The simulation results illustrate that WDMC is much more robust against bit errors.



(a) (b)

Figure 7. (a)KG-error of AFX-IC; (b)KG-error of WDMC.(**Cow** seuqnce under BER of 10^{-5})

We have to point out that there is still a probability of error propagation in WDMC. In Fig.7, it is shown that WDMC could recover from bit error and the probability for decoding failure is much less than AFX-IC. However, there is still error propagation observed. To make WDMC feasible for practical utility, error concealment scheme should be incorporated, while no such scheme is involved for fair comparison in our experiment.

3.3 Layered Coding and Progressive Coding

Based on layered SWC,[24] WDMC can compress and transmit mesh frames on different quality according to the decoder's requirement. This attribute is similar to PSNR scalability in video coding. In Fig.8, two least significant bit planes of the left mesh frame are not transmitted. For the right one, six least significant bit planes are not transmitted. Hence, the equivalent quantization step-size of the right one is $2^4 = 16$ times of the left one and costs less bits. Different levels of noise introduced by quantization are clearly observed on the face.

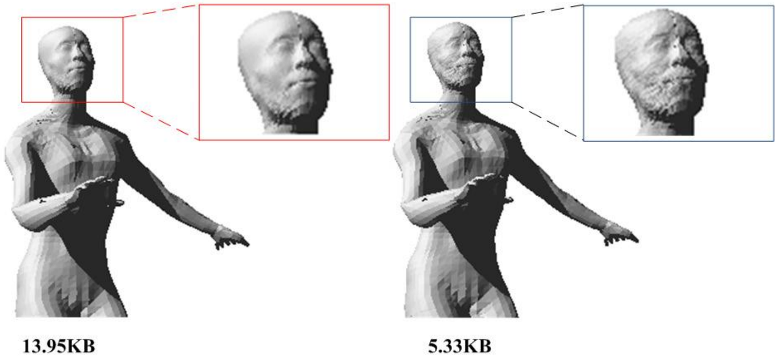


Figure 8. layered coding

Based on anisotropic wavelet transform, our codec can be extended to support progressive coding. This attribute is similar to resolution scalability in video coding. In streaming applications of 3D meshes, different users have different requirements for rendering. For instance, a mobile phone user doesn't need to render meshes in very fine resolution while a CAD software user would like to inspect his mesh model with very high resolution. In this sense, progressive coding optimally allocates network resource for different users.

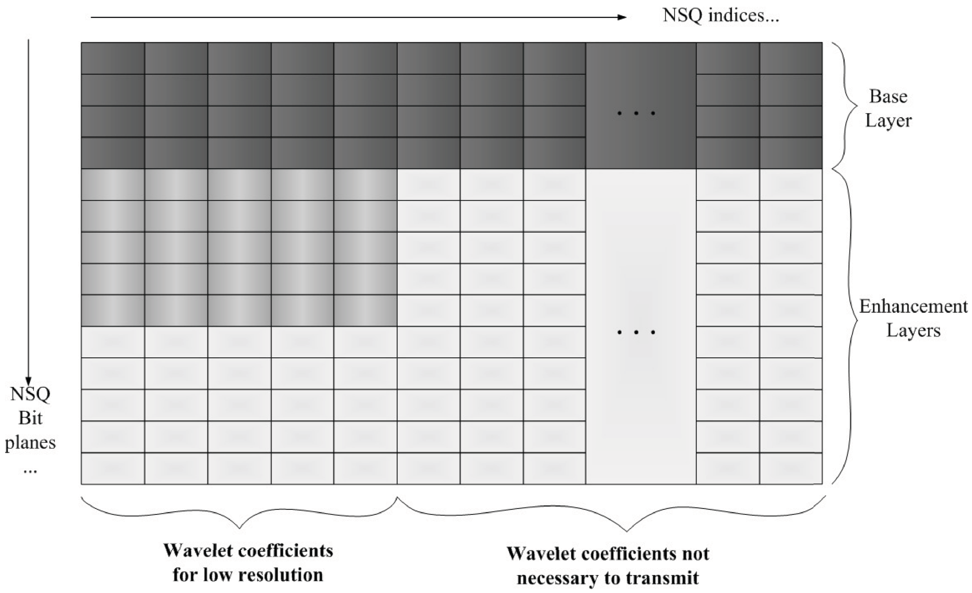

NSQ indices...
Base Layer
Enhancement Layers
NSQ Bit planes ...
. . .
. . .
Wavelet coefficients for low resolution
Wavelet coefficients not necessary to transmit

Figure 9. streaming scheme in progressive coding

To support progressive coding, we compress several most significant bit planes with SWC as base layer and the rest of the bit planes serve as enhancement layers which are transmitted without compression. Fig.9 shows our streaming scheme of progressive coding.

Within the enhancement layers, only the bits corresponding to required resolution is transmitted. The other bits need not to be transmitted. Fig.10 shows an example of progressive coding using **Dance** sequence. Twelve most significant bit planes are encoded as base layer. The enhancement layer consists of two bit planes. But only the necessary bits for the required resolutions are transmitted.

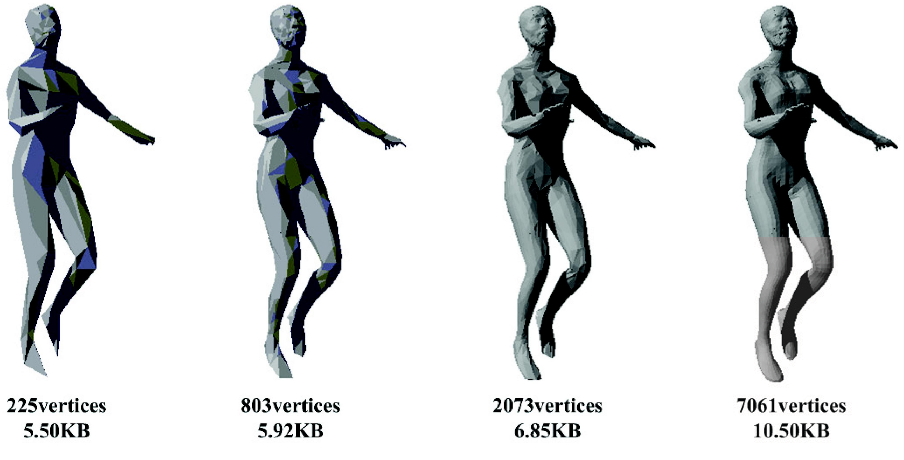


Figure 10. streaming scheme in progressive coding

Note that the enhancement layers are not compressed with SWC. That will result in some rate penalty when full resolution streaming is required by decoder. However, for low resolution transmission, our scheme is much more efficient. In conventional progressive mesh compression, the significance map of every coefficient is compressed as the base layer. Then the other bits are transmitted without compression as refinement information.[12, 34, 35] However, it is not suitable for our robustness-oriented codec design because critical structural information is contained in the significance map. Even a single bit error will cause decoding failure.

4. CONCLUSION AND LIMITATION

In this paper, a 3D dynamic mesh compression scheme based on Wyner-Ziv coding principles is proposed. Benefiting from the robustness of distributed source coding, the proposed codec manifests its robustness over conventional predictive coding schemes. Furthermore, based on anisotropic wavelet transform, the proposed scheme can be extended to support progressive coding which is a very desirable attribute for rendering and transmitting the dynamic mesh on different resolution. As for the compression performance, it is competitive to state of art standard approach.

5. ACKNOWLEDGEMENT

This work is supported by the key project of NSFC(No.60432030) and the Distinguished Young Scholars of NSFC(No.60525111). We would like to thank Igor Guskov from University of Michigan for providing us source code about 28 and the dynamic 3-D mesh sequences.

REFERENCES

1. J. Peng, C.-S. Kim, and C.-C. Kuo, “Technologies for 3d mesh compression: A survey,” *Journal of Visual Communication and Image Representation* **16**, pp. 688–733, 2005.
2. J. Lengyel, “Compression of time dependent geometry,” *ACM 1999 Symposium on Interactive 3D Graphics*, 1999.
3. 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. Circuits Syst. Video Technol* **12**, pp. 1178–1184, 2002.
4. L. Ibarria and J. Rossignac, “Dynapack: space-time compression of the 3d animations of triangle meshes with fixed connectivity,” *Proceedings of the 2003 ACM SIGGRAPH / Eurographics Symposium on Computer Animation*, pp. 126–135, 2003.
5. J.Zhang and C.B.Owen, “Octree-based animated geometry compression,” *Data Compression Conference* , pp. 508–517, 2004.
6. I. JTC1/SC29/WG11, *Information technology - Coding of Audio-Visual Objects. Part 2: Visual: 2001 Edition*, Doc. N4350, Sydney,Australia, 2001.
7. K. Mller, A. Smolic, M. Kautzner, P. Eisert, , and T. Wiegand, “Predictive compression of dynamic 3d meshes,” *Proc. ICIP2005, IEEE International Conference on Image Processing* , 2005.
8. M. Alexa and W. Muller, “Representation animations by principle components,” *EUROGRAPHICS 2000* **19**, pp. 291–301, 2004.
9. Z. Karni and C. Gotsman, “Compression of soft-body animation sequences,” *Computers & Graphics, Special Issue on Compression* **28**, pp. 25–34, 2004.
10. H. Briceno, P. Sander, L. McMillan, S. Gortler, and H. Hoppe, “Geometry videos: a new representation for 3d animations,” *Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer animation*, pp. 136–146, 2003.
11. F. Payan and M. Antonini, “Wavelet-based compression of 3d mesh sequences,” *Proceedings of IEEE ACIDCA-ICMI’2005*, (Tozeur, Tunisia), 2005.
12. I. Guskov and A. Khodakovsky, “Wavelet compression of parametrically coherent mesh sequences,” *Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer animation*, pp. 183–192, 2004.
13. J. Cho, M. Kim, S. Valette, H. Jung, and R. Prost, “3-d dynamic mesh compression using wavelet-based multiresolution anylysis,” *Proceedings of 2006 IEEE International Conference on Image Processing (ICIP)*, pp. 529 – 532, 2006.
14. E. Jang, J.D.K.Kim, S. Y. Jung, M.-J. Han, S. O. Woo, and S.-J. Lee, “Interpolator data compression for mpeg-4 animation,” *IEEE Trans. Circuits Syst. Video Technol* **14**, pp. 989– 1008, 2004.
15. J. Yang, C. Kim, and S. Lee, “Semi-regular representation and progressive compression of 3-d dynamic mesh sequences,” *IEEE Transactions on Image Processing* **15**, pp. 2531 – 2544, 2006.

16. J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory* **IT-19**, pp. 471–480, 1973.
17. A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory* **IT-22**, pp. 1–10, 1976.
18. R.Zamir, S.Shamai, and U.Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inf. Theory* **48**, pp. 1250–1276, 2002.
19. S.Pradhan, J.KUsuma, and K.Ramchandran, "Distributed compression in a dense microsensor network," *IEEE Signal Process Mag.* **19**, pp. 51–60, 2002.
20. Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Process Mag.* **21**, pp. 80–94, 2004.
21. B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE* **93**, pp. 71–82, 2005.
22. R.Puri, A.Majumdar, and K.Ramchandran, "Prism: A video coding paradigm with motion estimation at the decoder," *IEEE Transactions on Image Processing* **16**, pp. 2436 – 2448, 2007.
23. Q. Xu and Z. Xiong, "Layered wyner-ziv video coding," *IEEE Transactions on Image Processing* **15**, pp. 3791 – 3803, 2006.
24. Q. Xu and Z. Xiong, "Successive refinement for the wyner-ziv problem and layered code design," *IEEE Transactions on Signal Processing* **53**, pp. 3269 – 3281, 2005.
25. L.A.Dalton, "Analysis of 1-d nested lattice quantization and slepian-wolf coding for wyner-ziv coding of i.i.d. sources," *Texas A&M Univ. College Station, TX, Tech.Rep.2003* .
26. M. Lounsbery, "Multiresolution analysis for surfaces of arbitrary topological type," *Ph.D thesis, Dept. of Computer Science and Engineering, U. of Washington* , 1994.
27. I.Daubechies, I.Guskov, P.Schroder, and W.Sweldens, "Wavelets on irregular point sets," *Philosophical Transaction on Royal Society London A* **357(1760)**, pp. 2397–2413, 1999.
28. I. Guskov, W. Sweldens, and P. Schroder, "Multiresolution signal processing for meshes," *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* , pp. 325–334, 1999.
29. H. Hoppe, "Progressive meshes," *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics* , pp. 99–108, 1996.
30. A.D.Liveris, Z. Xiong, and C.N.Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," *Proceedings of Data Compression Conference, 2003.* , pp. 193 – 202, 2003.
31. A.Aaron and B.Girod, "Compression with side information using turbo codes," *Proceedings of Data Compression Conference, 2002.* , pp. 252 – 261, 2002.
32. A.D.Liveris, Z. Xiong, and C.N.Georghiades, "Compression of binary sources with side information at the decoder using ldpc codes," *Communications Letters, IEEE* **6(10)**, pp. 440 – 442, 2002.
33. L.Bahl, J.Cocke, F.Jelinek, and J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory* **20(2)**, pp. 284 – 287, 1974.
34. L. Jiankun and C.-C.J.Kuo, "Progressive coding of 3-d graphic models," *Proceedings of the IEEE* **86(6)**, pp. 1052 – 1063, 1998.
35. A. Khodakovsky, P. Schroder, and W. Sweldens, "Progressive geometry compression," *Proceedings of the 27th annual conference on Computer graphics and interactive techniques* , pp. 271 – 278, 2000.