LATTICE VECTOR QUANTIZATION FOR NORMAL MESH GEOMETRY CODING

Leonardo Hidd Fonteles and Marc Antonini

I3S laboratory, CNRS and University of Nice-Sophia Antipolis
2000, route des Lucioles, B.P. 121, 06903 Sophia-Antipolis Cedex, France
E-mail: fonteles@i3s.unice.fr, am@i3s.unice.fr

ABSTRACT

Multiresolution representation of surface meshes is known to be a powerful tool for modeling complex 3D objects. Among the existing schemes, normal meshes have proven to be very attractive for multiresolution and wavelet coding. However, most of the coding methods proposed in the literature are based on scalar quantization despite that vector quantization is known to be more efficient. In this work we propose a novel compression scheme based on a lattice vector quantizer which exploits the correlation inside each geometry subband in a multiresolution framework. Furthermore, we developed a model-based bit allocation algorithm able to work whatever the quantizer is and especially with vector quantization. The proposed scheme allows an improvement up to 1dB compared to the best state-of-the-art method.

1. INTRODUCTION

In computer graphics applications, triangular meshes have proven to be a powerful tool for the representation of 3D object surfaces. The demand for realism in graphics applications is such that today, 3D meshes can be defined by several millions of vertices, and more [1]. A simple representation of these highly detailed meshes becomes consequently huge and thus, interest has surged in recent years for developing 3D compression systems. Compression is a relevant solution to allow a compact storage or a fast transmission in bandwidth-limited applications.

Basically, triangle meshes are composed by two components: the vertex data and the connectivity between the vertices. The problem of connectivity compression has been well studied in the past and today many existing methods achieve good connectivity compression results [2]. Nowadays, more and more works consider the original mesh to be just one instance of the surface geometry and thus, consider the geometry to be the most important component of a mesh. In the case of geometry compression, the connectivity information is reduced to the minimum by remeshing the irregular input mesh using semi-regular remeshers [3, 4]. Among the existing schemes of semi-regular remeshing, the normal meshes [5] are attractive for wavelet coding, particularly with the unlifted butterfly wavelet transform [6].

Several wavelet coders exploit the normal meshes combined with the unlifted butterfly wavelet transform. Let us cite for example the normal mesh compression (NMC) proposed by Khodakovsky and Guskov based on a zerotree coder [6], the works of Lavu *et al* on estimation-quantization for mesh compression (EQMC), that exploits the spatial and inter-scale correlations of the normal meshes [7] and the works of Payan *et al* [8] optimizing the bit allocation across the wavelet subbands.

This work has been supported in part by Alβan Office.

All of these works are basically based on scalar quantization of the geometry. A recent work by Chou *et al* [9] was dedicated to vector quantization (VQ) of the vertex positions of the original irregular 3D object. VQ is a powerful tool extensively studied in audio, image and video coding. The work of [9] showed that VQ is also a promising tool for compressing the vertices of 3D triangle meshes.

In this paper we propose the use of VQ for lossy compression of geometry data in the framework of multiresolution analysis. Our scheme is based on lattice vector quantization (LVQ) designed for geometry coding of normal meshes and on optimal model-based bit allocation across the wavelet coefficient subbands. LVQ has proven to be a low complexity and robust method for vector quantization. Furthermore, the proposed bit allocation solves the problem of the estimation of rate-distortion functions and provides a simple and powerful tool for allocating binary resources at low cost and whatever the quantizer is.

This paper is organized as follows. In section 2 we present the overall proposed coding scheme. In section 3 we introduce LVQ and its adaptation to normal meshes. Section 4 deals with the proposed resource allocation and finally section 5 shows experimental results. We conclude in section 6.

2. OVERALL CODING SCHEME

The structure used for coding normal mesh objects follows the conventional transform-quantization-encoding paradigm. Fig. 1 presents the global scheme of the proposed coder. The algorithm principle is described hereinafter.

The normal remesher provides a semi-regular mesh, from the irregular input one. Then, a N-level unlifted butterfly wavelet transform [10, 11] is applied to obtain N subbands of three-dimensional wavelet coefficients. Using this wavelet transform ensures that most of the wavelet coefficients remains in the normal direction [6].

The quantization of each wavelet coefficient subband is ensured by a LVQ of dimension n (in this work $n = 3$) scaled by a scaling factor γ or equivalently a quantization step. Optimal quantization steps are chosen by a model-based bit allocation process across the different wavelet subbands (see section 4). To adapt the lattice to the source, we chose to represent each lattice vector by a product code depending on the statistic of the source to be quantized as proposed by Fischer in [12]. Clearly, if the source is Laplacian the product code consists of a prefix corresponding to the L_1 norm of a vector and the suffix to its position on the hyperpyramid with radius equal to the correponding L_1 norm.

The prefix and the suffix of each vector are then independently encoded by a stack-run encoder followed by an arithmetic coder [13]. See Section 3.3 for more details.

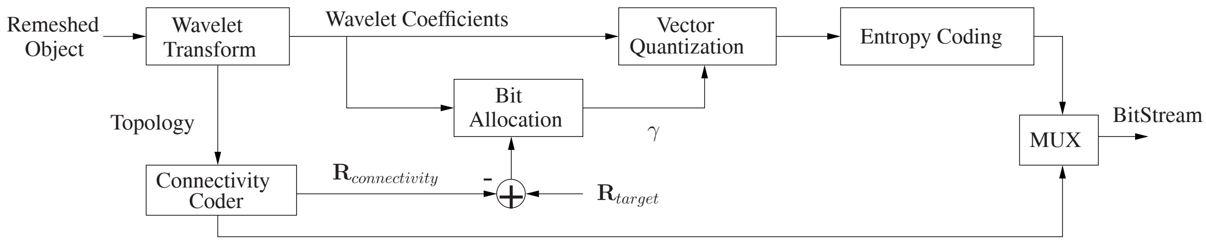

Fig. 1. Global coding scheme. The entropy coding box correspond to a stack-run coder followed by arithmetic coding.

The topology information is encoded using a connectivity coder like [2].

3. LVQ FOR NORMAL MESH

In this section we describe the lattice vector quantization principle and its adaptation to 3D normal remeshed objects.

3.1. Background

In data compression context, considerable attention has been given to the quantization of generalized gaussian distributed vectors by means of pyramidal or spherical lattice vector quantizer [12]. Such quantization is performed to an integer lattice lying on a pyramidal or spherical shell. A lattice Λ in $\mathbb{R}^n$ is composed of all integral combination of a set of linearly independent vectors $\mathbf{a}_i$ (the basis of the lattice) such that:

$$\Lambda = \{\mathbf{x} | \mathbf{x} = u_1\mathbf{a}_1 + u_2\mathbf{a}_2 + ...u_n\mathbf{a}_n\} \tag{1}$$

where the u_i are integers.

The fundamental advantage of lattice quantization is that no codebook needs to be generated or stored and quantization is very fast because it does not depend on the number of codewords used. Furthermore, encoding can be done using a prefix code [12], which is well suited to the regular structure of the lattice. The prefix code decomposes a vector $\mathbf{v}$ into its p norm $\|\mathbf{v}\|_p$ and an index number I, which is given by an algorithm of enumeration, such as [14].

Among the different lattices, the Z_n is known to give the best performance for low bit-rate quantization of generalized gaussian sources when p is small [15]. This is due to the better adaptation of the Z_n voronoi regions to the anisotropicity of the generalized gaussian sources probability density function (pdf), which is main concentrated along with the coordinate axes [15]. This is exactly the same context of the high frequency subband of a normal remeshed object, which is often coded at low bit-rate (see Section 4.2)

In this work we use LVQ based on the Z_n lattice, including dead-zone.

3.2. Proposed LVQ for normal meshes

In the case of normal meshes, and when the wavelet coefficients are expressed according to a system of local frames depending on the coarser mesh, a lot of coefficients have indeed small tangential components. Consequently, the most significant geometry information lies in the normal components [5]. Therefore, in order to take into account efficiently both normal and tangential informations we propose to adapt the LVQ structure.

Let us introduce an angular thresholding which defines a cone of angle Θ with the normal. For all the vectors $\mathbf{v}$ located inside this cone we force their tangential component to zero, as shown in Fig. 2. The thresholded vector is now a vector of the form $\mathbf{v}_{in} = (0, 0, z)$. The vectors $\mathbf{v}_{out} = (x, y, z)$ lying outside of the cone remain unchanged.

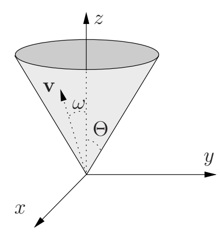

Fig. 2. Angular thresholding

This angular thresholding is implemented through an algorithm which is proposed as follows:

1. if $tan^{-1}\left(\frac{\sqrt{x^2+y^2}}{|z|}\right) < \Theta$
2. then, norm $= z$ and index position $= 0$;
3. else, norm $= \|\mathbf{v}_{out}\|_p$ and index position $=$ I;

Instead of coding the norm of the vector lying inside the cone, we actually code its normal component, and its index position is set to zero. For vectors lying outside the cone, a classical product code is used.

As a result, a non-null vector will present as many zeros as we enlarge Θ, involving a small impact on the quality of the reconstructed mesh, since small values of Θ are sufficient for a good thresholding (see Section 5).

3.3. Stack-Run for prefix code

Stack-run [13] is a variable length coding method developed to deal with integer numbers. These numbers are divided in two classes of values: zeros and significant numbers. The significant values (stacks) are coded directly into the binary form, using the symbols '0' and '1' for all bits, except for the most significant one, which is substituted by the '-' or '+' symbols according to its sign. On the other hand, the zeros are coded jointly into runs, where the number of zeros of a run is written in binary form, using the symbols '-' and '+' instead of '0' and '1', respectively.

In our framework, we need to code two sequences of integer numbers: one containing all the norms of the thresholded vectors and the other containing their corresponding indices. As said before, the indices sequence presents an increasing number of zero values according to the angular threshold Θ (great value of Θ involves high number of zeros). In that case, the stack-run coder will be efficient.

Furthermore, the stack-run coder will also be efficient for the sequence of norms, especially for high frequency subbands where a lot of vectors will be quantized by zero (see Section 4.2).

Then, an arithmetic coder is used to encode the two different sequences [13].

4. RESOURCE ALLOCATION

The overall coding procedure is performed independently for each subband. Then, a resource allocation procedure is necessary in order to distribute in an optimal way the amount of binary resources across each subband, allowing the best choice of scaling factor γ to each one. This allocation process is described in next sections.

4.1. Allocation principle

The coding process of each subband i requires an amount of R_i bits per semi-regular vertex. The overall bit-rate $\mathbf{R}$ used to code the entire object is computed as:

$$\mathbf{R} = \sum_{i=1}^{M} a_i R_i \tag{2}$$

where a_i is the fraction of the number of vertex in the i-th subband and the total number of vertex of all M subbands. This total bit-rate should be smaller than a target value $\mathbf{R}_{\max}$, imposed for coding the quantized coefficients (from Fig. 1, $\mathbf{R}_{\max}$ is $\mathbf{R}_{target} - \mathbf{R}_{connectivity}$). This means that:

$$\sum_{i=1}^{M} a_i R_i \leq \mathbf{R}_{\max} \tag{3}$$

In this context, the rate allocation problem consists in finding $\mathbf{R}$ minimizing the following distortion criterion:

$$D(\mathbf{R}) = \sum_{i=1}^{N} w_i D(R_i) \tag{4}$$

under the constraint given by (3). The weights w_i take into account the non orthogonality of the Butterfly wavelet filters [8]. To solve this problem, we use a Lagrangian approach and introduce the following Lagrangian functional $J(\mathbf{R}, \lambda)$:

$$J(\mathbf{R}, \lambda) = \sum_{i=1}^{M} w_i D_i(R_i) - \lambda(\sum_{i=1}^{M} a_i R_i - R_{\max}) \tag{5}$$

By imposing the zero-gradient condition, we find that the resulting optimal rate allocation vector $\mathbf{R}^* = \{R_i^*\}_{i=1}^{M}$ verifies the following set of equations:

$$\frac{w_i}{a_i}\frac{\partial D_i}{\partial R_i}(R_i^*) = \lambda \qquad \forall i \in \{1, \ldots, M\} \tag{6}$$

where λ is the Lagrange multiplier. We can read (6) this way: the optimal rates correspond to points having the same slope on the "weighted" curves $(R_i, \frac{w_i}{a_i} D_i)$. Note that $\lambda < 0$ since the RD curve are strictly decreasing. A simple dichotomic search algorithm on λ [16] is used to find the optimal rate $\mathbf{R}$ satisfying equations (6) and (3).

4.2. Model for $D(\mathbf{R})$ curves

The algorithm presented in section 4.1 not only requires the knowledge of the RD curve of each subband, but also supposes that these curves are differentiable, convex and accurate enough. The estimation of the RD curves is thus a crucial step. We propose to use in this paper the method prosed by [16]. This approach can be seen as both data-driven and model-based. It combines the advantages of algorithms based on RD curves analysis, and of model-based algorithms. Indeed, for each subband we evaluate experimentally (R,D) points located on the convex hull of the RD function and distributed between an ad hoc range of bitrates[1]. Then, these real RD curves are modeled using smoothing B-Splines [17] providing a continuous, convex and differentiable analytical model for each RD function of each subband (see a typical example in Fig. 3).

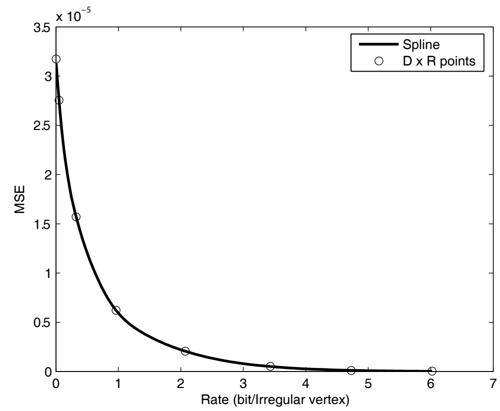

Fig. 3. Modeling of a typical RD curve using smoothing B-Spline

5. EXPERIMENTAL RESULTS

In this section we evaluate the performance of the proposed coder. We compare its results to the state-of-the-art coders, namely the Normal Mesh Coder (NMC) [6] and the Estimation-Quantization geometry coder for normal meshes (EQMC) [7].

We use the Peak-Signal-to-Noise-Ratio (PSNR), given in decibels (dB), to evaluate the performances of the proposed method. This value is calculated as it follows:

$$PSNR = 20 \log_{10}\left(\frac{BB}{d_S}\right) \tag{7}$$

where *BB* represents the diagonal length of the bounding box of the object and d_S the *surface-surface distance* between the semi-regular coded object and the original irregular one. Both parameters are computed using the Mesh algorithm [18].

The figures 4 and 5 show results for the Venus and Horse normal remeshed objects[2]. In both case, the angular threshold Θ is set to 1^{o} (simulations show that it is a good value for all tested objects), and codebooks are designed for Laplacian distribution. The proposed encoder gives good compression results and a gain up to 1dB in comparison with the two state-of-the-art methods.

[1]The range depends on the bandwith of the subband, i.e., for high frequency subbands small bitrates will be allocated while for low frequency subband high bitrates will be allocated.

[2]The Horse and Venus remeshed models are courtesy of Multi-Res Modeling Group http://www.multires.caltech.edu

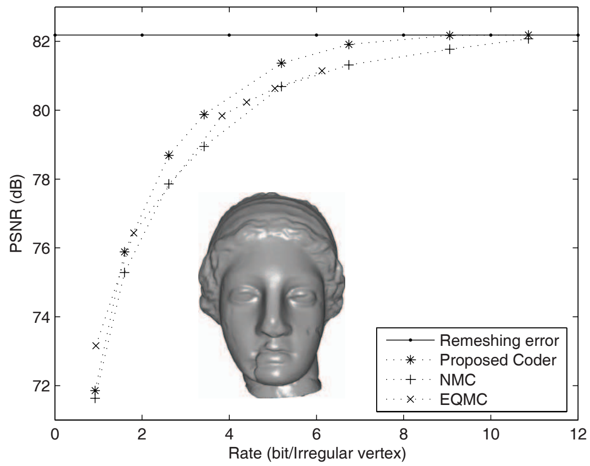

Fig. 4. PSNR vs Rate curves for the proposed coder and the state-of-the-art for Venus object

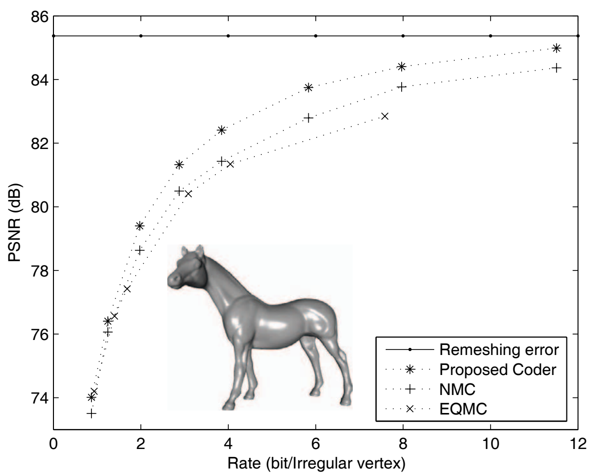

Fig. 5. PSNR vs Rate curves for the proposed coder and the state-of-the-art for Horse object

6. CONCLUSION

In this paper we have shown that LVQ associated to a stack-run coder is promising for the compression of normal mesh objects. Furthermore, the introduction of a model-based bitrate allocation using spline is well suited to vector quantization. The experimental results showed that combining optimal spline-based bit allocation and LVQ gives good compression results with gain up to 1dB in comparison with state-of-the-art methods.

In future works, we propose to design LQV for vectors with higher dimension, by taking into account the spatial correlation between vertices.

7. REFERENCES

[1] M. Levoy, "The digital michelangelo project," in *Proceedings of the 2nd International Conference on 3D Digital Imaging and Modeling*, October 1999.

[2] C. Touma and C. Gotsman, "Triangle mesh compression," *Graphics Interface'98*, pp. 26–34, 1998.

[3] Pierre Alliez and Craig Gotsman, *Recent Advances in Compression of 3*D *Meshes*, 2005.

[4] Aaron Lee, Henry Moreton, and Hugues Hoppe, "Displaced subdivision surfaces," in *Proceedings of ACM SIGGRAPH 2000*, July 2000, pp. 85–94.

[5] Igor Guskov, Kiril Vidimce, Wim Sweldens, and Peter Schröder, "Normal meshes," in *SIGGRAPH 2000, Computer Graphics Proceedings*, 2000, pp. 95–102.

[6] A. Khodakovsky and I. Guskov, "Normal mesh compression," Geometric Modeling for Scientific Visualization, Springer-Verlag, 2002.

[7] S. Lavu, H. Choi, and R. Baraniuk, "Geometry compression of normal meshes using rate-distortion algorithms," in *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry processing*, 2003.

[8] F. Payan and M. Antonini, "An efficient bit allocation for compressing normal meshes with an error-driven quantization," *Computer Aided Geometric Design, Special Issue on Geometric Mesh Processing*, vol. 22, pp. 466–486, July 2005.

[9] Peter H. Chou and Teresa H. Meng, "Vertex data compression through vector quantization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 4, pp. 373–382, 2002.

[10] Wim Sweldens, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998.

[11] P. Schröder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," *Proceedings of SIGGRAPH 95*, pp. 161–172, 1995.

[12] T. R. Fischer, "A pyramid vector quantizer," *IEEE Trans. Inform. Theory*, vol. 32, pp. 568–583, July 1986.

[13] M.J. Tsai, J. Villasenor, and F. Chen, "Stack-run coding for low bit rate image communication," in *Proceedings of the IEEE International Conference on Image Processing*, Lausanne, Switzerland, september 1996, vol. 1, pp. 681–684.

[14] P. Loyer, J.M. Moureaux, and M. Antonini, "Lattice codebook enumeration for generalized gaussian source," *IEEE Transactions on Information Theory*, vol. 49, pp. 521–528.

[15] Z. Gao, F. Chen, B. Belzer, and J. Villasenor, "A comparison of the Z, E8, and leech lattices for quantization of low-shape-parameter generalized gaussian sources," *IEEE Signal Processing Letters*, vol. 2, pp. 197–199, October 1995.

[16] M. Cagnazzo, T. André, M. Antonini, and M. Barlaud, "A model-based motion compensated video coder with jpeg2000 compatibility," in *Proceedings of IEEE International Conference on Image Processing*, Singapore, October 2004, pp. 2255–2258.

[17] M. Unser, "Splines: A perfect fit for signal and image processing," *IEEE Signal Processing Magazine*, vol. 16, no. 6, pp. 22–38, november 1999.

[18] N. Aspert, D. Santa-Cruz, and T. Ebrahimi, "Mesh: measuring erros between surfaces using the hausdorff distance," in *Proceedings of IEEE International Conference on Multimedia and Expo*, 2002, vol. 1, pp. 705–708.