An Algebraic Multi-Grid Approach for High-Pass Quantization

Ofir Engolz Rony Goldenthal Dani Lischinski Daniel Cohen-Or

The Hebrew University of Jerusalem Tel-Aviv University

Abstract

Today's huge irregular polygon meshes require effective compression techniques to reduce the associated storage requirements, network bandwidth and transmission times. In this paper, we describe a new method for compressing the geometry of an irregular triangle mesh, which is both scalable (encoding and decoding time is linear in the number of mesh vertices) and progressive, enabling the decoder to generate a progression of approximations while the data is being transmitted. The encoding approach utilized by our method is based on the recently introduced High-Pass Quantization technique. The decoding stage, however, uses a linear Algebraic Multi-Grid solver, which makes high-pass quantization truly scalable. The speed of our solver makes it possible to update the solution as more bits are received, resulting in a progressive transmission scheme.

CR Descriptors: I.3.5 [**Computer Graphics**]: Computational Geometry and Object Modeling — *surface, solid, and object representations*

1 Introduction

Irregular triangle meshes are commonly used for representing 3D objects of arbitrary topology in a wide variety of application areas, ranging from medical and scientific visualization to cultural and historical preservation. With the advent of range scanning technologies in the last decade, these meshes are often very detailed, containing hundreds of thousands and even millions of vertices. Thus, working with such meshes requires effective compression techniques to reduce storage requirements and transmission times. Progressive transmission techniques are particularly desirable, since they drastically reduce the apparent latency experienced by a user downloading such a mesh.

In this paper we develop a method for compressing the geometry of an irregular triangle mesh, which is both scalable and progressive. Specifically, we build upon the high-pass quantization (HPQ) method for mesh encoding recently introduced by Sorkine *et al.* [SCOT03]. This method encodes the geometric coordinates of the mesh vertices by applying a transformation based on the mesh Laplacian and then quantizing the transformed coordinates. As demonstrated by Sorkine *et al.*, such an approach allows considerably more aggressive quantization to be applied without introducing significant visual errors, since the quantization error is mostly comprised of low-frequency bands.

However, the decoding stage of the HPQ method invloves solving a large linear least squares problem, and the solvers used by Sorkine *et al.* are super-linear and thus are only practical for moderately sized meshes (under 100,000 vertices). In this work we advocate using an Algebraic Multi-Grid (AMG) solver, which exhibits roughly linear performance, and thus makes the HPQ method truly scalable. In fact, this solver is sufficiently fast to update the solution as more bits are received, resulting in a progressive transmission scheme.

Although multigrid methods have long been recognized as the most efficient and appropriate solvers for a wide variety of large linear problems [TOS01], their application to problems in digital geometry processing has been limited so far. A few exceptions [RL03; AKS04] are discussed in the next section. In this work we chose to use the Algebraic Multi-Grid (AMG) method, since it does not rely on the geometry of the mesh, and thus enables us to set up the solver on the receiving end before the geometry is available.

2 Previous Work

Mesh compression approaches must address two conceptually separate problems: connectivity encoding and geometry encoding. State-of-the-art connectivity encoders are very effective, typically requiring between 2 to 6 bits per vertex [TG98; COLR99; GS98; KADS02]. In this work we make use of such an encoder in order to encode and transmit the connectivity information before geometry transmission commences.

The availability of near-optimal connectivity encoders emphasizes the need for effective geometry compression and encoding schemes. Standard linear predictor based geometry encoders [TG98] result in encodings where the geometry is, on average, at least five time larger than the connectivity. Recent compression methods represent the mesh geometry using special bases: Karni and Gotsman [KG00] use a spectral basis, inspired by the Fourier basis; Khodakovsky *et al.* [KSS00] used a wavelet basis. While the latter method is able to achieve very good compression ratios, it requires a semi-regular remeshing of the input mesh, which is undesirable in some applications.

As mentioned earlier our method is based on the HPQ method for encoding the geometry of irregular meshes [SCOT03]. Like Karni and Gotsman's method, HPQ preserves the original connectivity of the mesh and is based

on the spectral properties of the mesh Laplacian. However, while Karni and Gotsman introduce errors into the high frequency modes, in the HPQ method the errors are mostly confined in the low frequency modes. The HPQ method is explained in more detail in the next section.

In this work, we introduce a scalable and progressive variant of HPQ by utilizing an AMG solver. Multi-grid solvers for digital geometry processing have been previously explored by Aksoylu *et al.* [AKS04], who compared several different mesh coarsification strategies for constructing mesh hierarchies suitable for multi-level solvers. The mesh hierarchy used in this work is similar to their fast decaying *MIS hierarchy*.

Ray and Levy [RL03] utilize a cascadic multi-grid solver for speeding up the computation of least squares conformal maps. In this work we use a more general AMG method, and achieve even more substantial speedups.

3 High-Pass Quantization

Let M be a triangular mesh with a set of n vertices whose Cartesian coordinates in R^3 are denoted as $v_i = (x_i, y_i, z_i)$. Rather than quantizing these cartesian coordinates directly, HPQ first transforms them into *relative* or δ*-coordinates*, which encode the difference of each vertex i from the center of mass of its ring on mesh neighbors $Star(i)$:

$$\delta_i = (\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)}) = v_i - \frac{1}{d_i} \sum_{k \in Star(i)} v_k \qquad (1)$$

where d_i is the degree of vertex i. This transformation is conveniently expressed in matrix form as

$$L\vec{v} = D\vec{\delta} \qquad (2)$$

where D is the diagonal matrix $D_{i,i} = d_i$ and L is the mesh Laplacian matrix [Fie73]:

$$L_{i,j} = \begin{cases} d_i & i == j \\ -1 & j \in Star(i) \\ 0 & \text{otherwise.} \end{cases}$$

The HPQ method encodes the geometry of the mesh by quantizing the δ-coordinates, followed by standard entropy encoding. Thus, in order to decode the geometry it is necessary to solve the linear systems (2) with the quantized version of $\vec{\delta}$ on the right-hand side. However, the Laplacian matrix L is singular (since all mesh translations result in the same set of relative cooordinates). To overcome this problem and to reduce the errors introduced by quantizing $\vec{\delta}$, Sorkine *et al.* [SCOT03] *anchor* a small portion of the vertices to their original Cartesian coordinates. Specifically, to anchor the i-th vertex, Sorkine *et al.* append the row vector e_i^T (the i-th row of the identity matrix) to the Laplacian matrix and the original coordinates v_i to the right-hand side vectors. Having anchored k vertices in this manner, the resulting rectangular system is solved as a least-squares problem.

In this paper, we explore a different way of handling anchored vertices. Instead of appending additional rows to the Laplacian matrix, to anchor the i-th vertex we add the row vector αe_i^T to the i-th row of the matrix, and also add αv_i to the i-th entry of the right-hand side vectors. Informally, the i-th equation says that the corresponding vertex should satisfy both the Laplacian relationship with its neighbors and the anchoring constraint. The scaling factor α enables us to control the weight given to the anchoring constraint. Empirically, we have found values of $\alpha = 0.2$ to 0.3 to produce the most visually pleasing solutions for all of the models we experimented with. For values of $\alpha < 0.2$ the anchoring was not effective, while for values of $\alpha > 0.5$ the resulting solution was not sufficiently smooth. It should be noted that our solution is not equivalent to the least-squares approach employed by Sorkine *et al.*, as it does not guarantee that the anchors are optimally approximated in the least-squares sense. However, with our approach the matrix remains square, and our experiments indicate that the results produced by our approach are equally accurate (both visually and numerically) to those of Sorkine *et al.*

3.1 Non-uniform quantization

Instead of quantizing the relative coordinates vector $\vec{\delta}$ uniformly using the same number of bits for each vertex, like done by Sorkine *et al.*, we use non-uniform quantization. Vertices with large δ-coordinates suggest that they are more significant in the mesh (sharp points and corners), so we prefer to encode them more accurately. Thus, in order to quantize with the average rate of p bits per coordinate, after normalizing the $\vec{\delta}$ vector we use $p-1$ bits to quantizate the lower 10% of the δ-coordinates and $p+1$ bits to quantizate the top 10%. The remaining (non-anchor) vertices are quantized using p bits.

4 An AMG Solver

To solve equation (2) we employ an Algebraic Multi-Grid (AMG) approach. The idea behind multigrid methods in general is quite simple: starting with the full problem (finest level) the solver performs several *error smoothing* iterations, then *restricts* the residual onto a coarser level, solves for it recursively employing increasingly coarser levels, and *interpolates* the result to update the solution at the finest level. Thus, in order to construct a specific multigrid solver one needs to specify the following components:

1. Coarsening procedure for constructing the hierarchy of levels;
2. Restriction (interpolation) operators, which are necessary for switching from fine to coarse (coarse to fine) level;

3. Error smoothing procedure to apply at each level.

Ordinary (geometric) multigrid methods take the geometry of the problem into account when constructing the hierarchy of levels. In this work, we employ the more general AMG approach, in which geometry is not taken into account by the coarsening process. This is necessary in our setting, since the geometry is what we are solving for! Our specific coarsening procedure and the construction of restriction and interpolation operators is described in the next section. As for the smoothing procedure, we use standard Gauss-Seidel relaxation.

4.1 Coarsening

Our coarsening procedure operates in a manner similar to the method used by Aksoylu *et al.* [AKS04] for their *MIS hierarchy* construction. The idea is to select a subset of vertices and remove them from the mesh yielding the next coarser level. However, when selecting the vertices to be removed one must account for the fact that we will need an interpolation operator in order to propagate the solution achieved at the coarser level back to the finer level. Thus, the vertices chosen for removal (F-vertices) should depend in some simple manner on the vertices chosen to remain in the coarser level (C-vertices). In our approach we require that each F-vertex is adjacent to (depends on) at least one C-vertex. We look for a maximal independent set of vertices to serve as C-vertices, and remove all of the other vertices from the mesh to obtain the next coarser level. The value at an F-vertex is linearly interpolated from all C-vertices adjacent to it, yielding a very simple interpolation (prolongation) operator, and the corresponding restriction operator is defined simply as the transpose of the prolongation operator.

Starting with a mesh in which all vertices are unmarked, we sweep over all of the vertices in the mesh. An unmarked vertex is marked as independent (C-vertex) if none of its neighbors are marked as independent, and marked as dependent (F-vertex) otherwise. Once all vertices in the mesh have been marked, the prolongation and restriction operators are constructed, and the F-vertices are removed. The actual removal may be performed by a sequence of edge contraction: each F-vertex is removed by contracting an edge connecting this vertex to a C-vertex. Figure 1 illustrates the mesh coarsening process: the mesh on the left is the finer mesh with the C-vertices marked as bold circles, while the mesh on the right shows the resulting coarsened mesh.

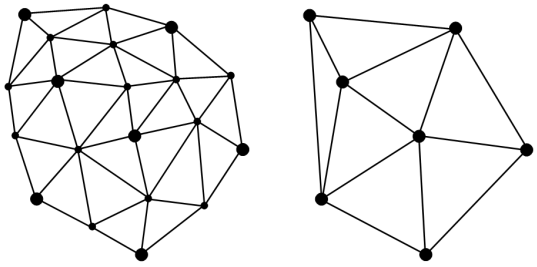

Figure 1: An illustration of the coarsening process. In the mesh on the left the independent vertices (C-vertices) are marked by bold circles. The right mesh contains only these vertices after all of the dependent vertices (F-vertices) have been removed.

5 Results

5.1 Scalability

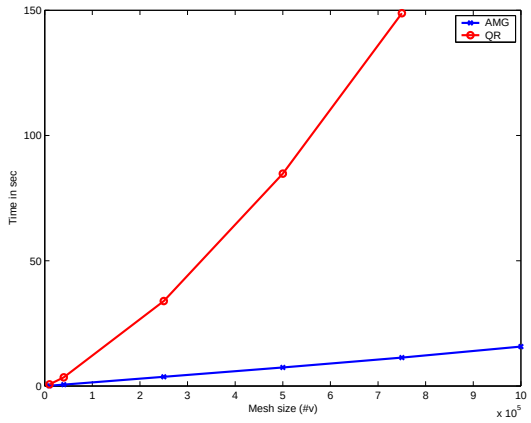

Figure 2: A comparison between AMG and QR decomposition. Running times (seconds on a Pentium 4 at 2.4GHz) are plotted as a function of the number of vertices.

We begin with a comparison bewteen our AMG solver and the QR-based solver used by Sorkine *et al.* [SCOT03]. To perform this comparison we measured the time each method required to decode a series of progressively more complex tessellations of the same object (a displaced plane). The corresponding timings are plotted in Figure 2. The plots clearly indicate that the AMG solver exhibits linear behvior, as predicted by theory, while the QR-based solver exhibits superlinear time complexity which makes it unpractical for complex meshes.

5.2 Progressivity

Using our method a progressive transmission scheme may be obtained as follows. The mesh connectivity information is encoded and transmitted first (using an state-of-the-art mesh connectivity encoder). Next an agressively quantized version of the geometry (e.g., 5-bits per coordinate on average) is encoded and transmitted. Next, a vector containing an additional bit for each coordinate is encoded and transmitted, and so forth until the desired encoding accuracy is met.

On the receiving (decoding) end, once the connectivity is received it is decoded and the coarsification process for con-

Quantization level	5 bits	7 bits	9 bits
Bits transmitted	5x19851	7x19851	9x19851
Time (sec)	1.36	1.81	2.26
Metro $E_{\max}$	0.0214	0.02218	0.000509
Metro E_{mean}	0.00636	0.000859	0.000107
Metro E_{RMS}	0.0076	0.000992	0.000132
$E_{\text{Haussdorff}}$	0.0214	0.002222	0.000509

Table 1: Transmitted geometry, reconstruction time, and errors for the progressive decoding of the Horse model. Errors were computed using the Metro tool, reported with respect to the bounding box diagonal.

Quantization level	5 bits	7 bits	9 bits
Bits transmitted	5x50000	7x50000	9x50000
Time (sec)	2.5	3.67	4.65
Metro $E_{\max}$	0.00236	0.000242	0.000044
Metro E_{mean}	0.000565	0.000074	0.000011
Metro E_{RMS}	0.000706	0.000088	0.000013
$E_{\text{Haussdorff}}$	0.002363	0.000242	0.000044

Table 2: Transmitted geometry, reconstruction time, and errors for the progressive decoding of the Venus model. Errors were computed using the Metro tool, reported with respect to the bounding box diagonal.

structing the multigrid hierarchy commences, since no geometry is required in this stage. Once the multigrid solver has been set up and the aggressively quantized geometry has been received the AMG solver is used to reconstruct the first crude approximation of the mesh geometry. As additional bits arrive, the AMG solver quickly updates the approximation resulting in a progression of approximation leading up to the final desired accuracy.

The outcome of this process is demonstrated in Figure 3. The leftmost images are the result of decoding the 5-bit quantized geometry, followed by 7-bit and 9-bit quantizations. For comparison, the original models are shown on the right. Tables 1 and 2 reports the amount of transmitted geometry bits, the cumulative reconstruction times, and the corresponding errors as computed by the Metro tool [CRS96]. Another example is shown in Figure 4.

6 Conclusions

In this paper we have developed a new method for decoding High-Pass Quantized geometry of irregular triangular meshes. We have shown that by utilizing a linear AMG solver we make high-pass quantization scalable and practical to much more complex meshes than was preiously possible. The fast solver was shown suitable for progressive transmission of high-pass quantized meshes.

Acknowledgement

Thanks go to Raanan Fattal for helpful discussion regarding multi-grid methods, and to Olga Sorkine and Sivan Toledo for helpful discussions and for providing us with their code.

References

[AKS04] Burak Aksoylu, Andrei Khodakovsky, and Peter Schröder. Multilevel solvers for unstructured surface meshes. *SIAM J. Sci. Comput.*, 2004. to appear.

[COLR99] Daniel Cohen-Or, David Levin, and Offir Remez. Progressive compression of arbitrary triangular meshes. In *Proceedings of the conference on Visualization '99*, pages 67–72. IEEE Computer Society Press, 1999.

[CRS96] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno. Metro: measuring error on simplified surfaces. Technical report, 1996.

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czech. Math. Journal*, 23(98):298–305, 1973.

[GS98] Stefan Gumhold and Wolfgang Straüer. Real time compression of triangle mesh connectivity. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 133–140. ACM Press, 1998.

[KADS02] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schröder. Near-optimal connectivity encoding of 2-manifold polygon meshes. *Graph. Models*, 64(3/4):147–168, 2002.

[KG00] Zachi Karni and Craig Gotsman. Spectral compression of mesh geometry. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co., 2000.

[KSS00] Andrei Khodakovsky, Peter Schröder, and Wim Sweldens. Progressive geometry compression. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 271–278. ACM Press/Addison-Wesley Publishing Co., 2000.

[RL03] Nicolas Ray and Bruno Levy. Hierarchical least squares conformal map. In *Proceedings of Pacific Graphics 2003*, 2003.

[SCOT03] Olga Sorkine, Daniel Cohen-Or, and Sivan Toledo. High-pass quantization for mesh encoding. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 42–51. Eurographics Association, 2003.

[TG98] C. Touma and Craig Gotsman. Triangle mesh compression. In *Proc. Graphics Interface '98*, pages 26–34, 1998.

[TOS01] U. Trottenberg, C. Oosterlee, and A. Schüller. *Multigrid*. Academic Press, 2001.

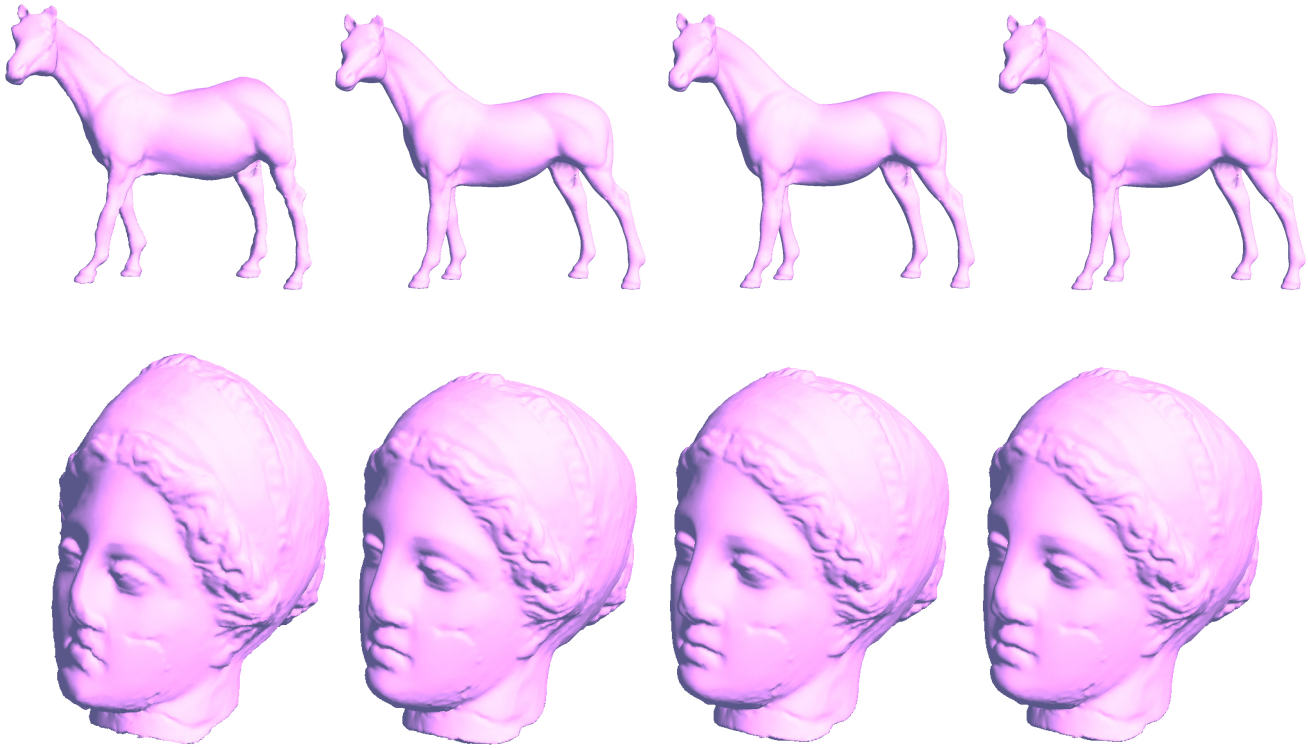

Figure 3: Progressive decoding of the Horse model (19,851 vertices, top row) and the Venus model (50,000 vertices, bottom row). From left to right: decoding the first 5 bits per coordinate, 7 bits, 9 bits, and the original model (shown here for reference). Low frequency errors make the differences difficult to perceive.

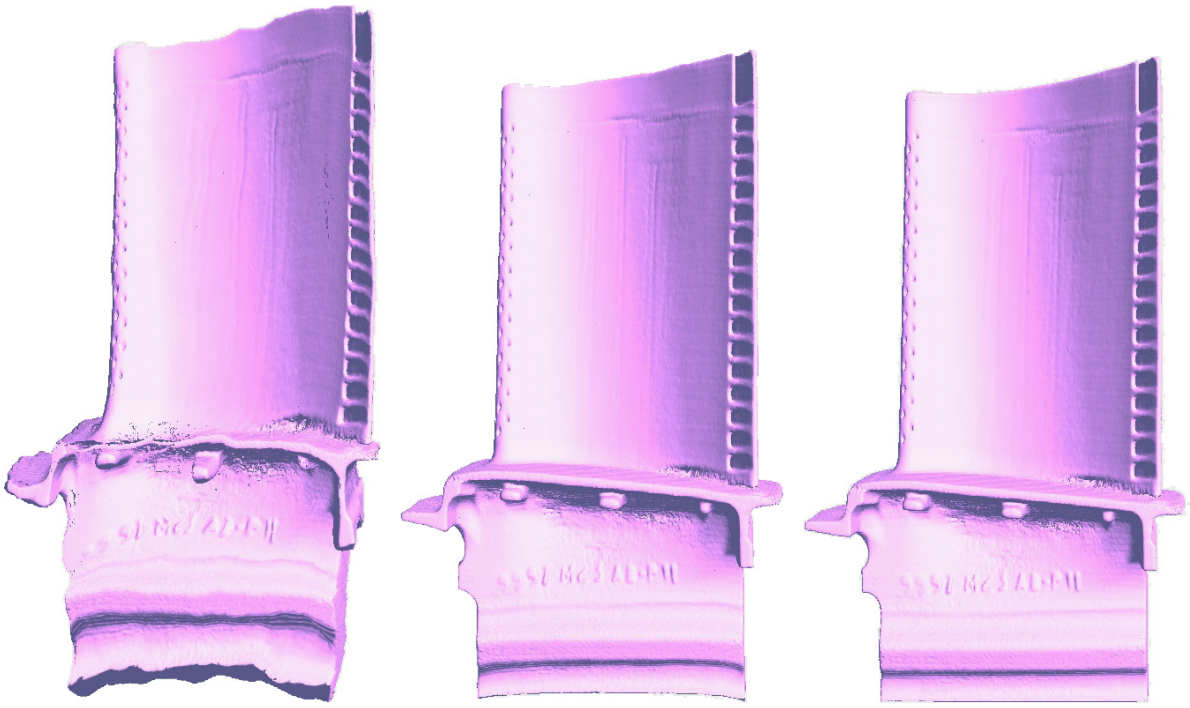

Figure 4: Blade model (882,954 vertices). From left to right: 7-bit quantization, 9-bit quantization, original mesh.