Hierarchical Shape-Adaptive Quantization for Geometry Compression

Stefan Gumhold

Max-Planck-Institute, AG4 Room 207
Stuhlsatzenhausweg 85,D-66123 Saarbrücken
Email:gumhold@mpi-sb.mpg.de

Abstract

The compression of polygonal mesh geometry is still an active field of research as in 3d no theoretical bounds are known. This work proposes a geometry coding method based on predictive coding. Instead of using the vertex to vertex distance as distortion measurement, an approximation to the Hausdorff-distance is used resulting in additional degrees of freedom. These are exploited by a new adaptive quantization approach, which is independent of the encoding order. The achieved compression rates are similar to those of entropy based optimization but with a significantly faster compression performance.

1 Introduction

In connectivity coding exist very efficient close to optimal coding techniques. Spectral coding [9] seems to be optimal for geometry compression, what has been shown for 2d by Ben-Chen and Gotsman [1]. The major disadvantage of spectral coding is its slow compression *and* decompression performance. Therefore are techniques based on predictive coding still very interesting and in most applications more

practical. Predictive coding has been introduced by Deering [2]. The predictor coefficients have been optimized by Taubin and Rossignac [13]. But it turned out to be more efficient to simply use a parallelogram prediction as proposed by Touma and Gotsman [14].

The parallelogram prediction is also used in this work. Geometry coding is directed by the connectivity encoder, for which the improved cut-border machine has been used [8, 4] but any other region growing scheme would work as well. The three vertex locations of the first encoded triangle are stored uncompressed. After that each newly encountered vertex is added together with a new triangle, which is edge-adjacent to an already processed triangle, whose vertex locations are known to encoder and decoder. The known triangle is called *reference triangle* and the edge connecting to the new triangle is called *gate*. The new vertex location is predicted by extending the reference triangle to a parallelogram. As both encoder and decoder can compute the predicted location, only the correction vector from the original vertex location to the prediction has to be encoded.

The parallelogram prediction has been improved in several ways: Lee and Ko [11] use vector quantization to encode the correction vectors. Kronrod and Gotsman [10] optimizing the traversal order of the connectivity encoder to minimize the length of the correction vectors. Gumhold and Amjoun [6] fit higher-order polygonal surfaces to improve prediction performance. In this work the simple parallelogram prediction is used. But instead of quantizing uniformly a new adaptive quantization strategy is proposed that bounds the Hausdorff-distance between original and quantized mesh. Although the proposed approach increases compression times noticeably, the decompression speed is not lowered at all and the compression rates are significantly better.

Two important issues of predictive coding are the coordinate system used to express the correction vectors and the entropy coding method used for coding the quantized vertex coordinates. Besides the commonly used world coordinate system, angular coordinates have been proposed by Lee et al. [12] and cylindrical by Gumhold and Amjoun [6]. In the proposed work the

local coordinate system is used that has been proposed by Lee and Ko [11]. The direction of the gate edge is identified with the x-axis and the normal of the reference triangle, which points to the exterior of the surface, is used as z-axis. The y-axis completes an orthonormal coordinate system. The use of the local coordinate system typically saves between one and three bits per vertex (*bpv*).

In section 4 an optimization scheme is proposed for the coding of the indices resulting from the shape adaptive quantization of the local coordinates. The method determines that best package size for the grouping of bits into adaptive probability models, which are used by the arithmetic coder. Although the scheme is very simple, it allows to save further one to three bpv, which has an important impact on the final compression rates.

In the following section the notation of validity region and shape adaptive quantization is introduced. Section 3 motivates and describes the proposed hierarchical quantization approach, which is independent of the encoding order. Results are given in section 5 before the paper is concluded with directions for future work.

2 Validity Regions

Adaptive quantization has first been proposed by Lee and Ko [11]. They collected the prediction correction vectors expressed in a local coordinate system. Then the LBG algorithm was applied to optimize a codebook of correction vectors. As the quantization step leads to error accumulation, not only the codebook but also additional correction bits needed to be transmitted in order to control the error.

Lee et al. [12] quantize adaptively by transforming the correction vectors into an angular coordinate system with a fixed quantization. No maximum error is guaranteed. Comparison to previous approaches is done with the root mean squared error (RMS). Also the use of validity regions was proposed in the work of Lee et al. A validity region is assigned to each of the vertices. It defines the region in space of all valid quantization locations around the vertex location. The quantizer then chooses from all valid quantization locations, the one that minimizes the current coding cost. An extensive sampling strategy was used to find the best quantization location. For each sampled location the entropy after coding was computed, what results in impractical compression times. Lee et al. only investigated rectangular regions although also quadrics were imagined; but they did not propose how to compute them and no implementation was reported. The results with the rectangular regions were disappointing and other more simple quantization schemes yielded better compression rates.

In this work the ideas of Lee et al. are developed further. We propose to define the validity regions from the Hausdorff distance between the original mesh $\mathcal{M}(\vec{p}_i)$ and the quantized mesh $\mathcal{M}(\vec{q}_i)$ with adaptively quantized vertex locations $\vec{q}_i$. To be able to compare the results with uniform quantization, the Hausdorff-distance was bound to

$$\epsilon_b = maxBBoxExtent/2^b,$$

where b is the number of quantization bits.

To realize the Hausdorff-bound for each vertex i of the to be encoded mesh a validity region R_i was constructed. The validity region R_i is defined as the union of all locations, onto which the vertex position $\vec{p}_i$ can be quantized without introducing a distortion larger than ϵ_b. As the explicit computation of the validity regions is complicated and as the regions are also affected by changes in the neighborhood, the R_i were approximated with ellipsoids in the following manner.

For each vertex a quadric error metric [3] was computed from the plane equations of the incident triangles and border edges. The iso-surfaces of the quadric error metric approximate the shape of the validity region. The iso-surfaces have larger extent in flat areas and smaller extent in curved areas. The validity region was set to the iso-surface of the quadric error metric whose smallest radius is ϵ_b. As the ellipsoids are oriented in tangential space, they do not follow a best fit curvature hyperboloid and we shrunk the validity regions by a safety factor of 0.6.

In order to avoid flipping of triangles in very flat areas, where the validity regions have a large extend, the maximum radius of the validity ellipsoid was limited by half of the length of the shortest edge adjacent to the corresponding vertex.

3 Adaptive Hierarchical Quantization

3.1 Overview

Figure 1 illustrates the proposed adaptive quantization approach. It shows the original mesh in light grey with darkly filled vertex locations. The validity region around the original vertex is the ellipse shaded in very light grey. The predicted location is shown as the vertex with inverted shading.

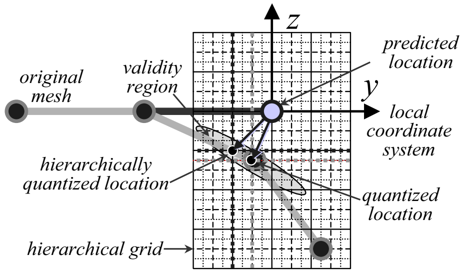


Figure 1: Illustration of the adaptive hierarchical quantization approach.

The hierarchical quantization grid is built around the predicted location in the local coordinate system. In uniform quantization the quantized location would be the small ball on the fatly dotted grid lines shaded in grey. In the adaptive quantization approach one can choose from all valid grid locations (inside the validity region) the one with the smallest coding cost.

In the next section the coding cost for the different grid locations will be examined. It turns out that every second grid line – the ones with even indices – save one coding unit, every fourth two units and so on. Remember that the counting of the grid lines starts with zero at the predicted location. In Figure 1 the most expensive grid lines are finely dotted, the ones that save one unit are dashed and the ones that save three units are solid. In 3d the saved units are summed over the three coordinate directions and the adaptive quantization location is chosen as the one with the largest sum. In Figure 1 it is the location of the small ball on the intersection of the fatly dotted grid lines in dark grey, for which 4 units can be saved.

3.2 Quantization Approaches

Four approaches have been implemented to find the optimal adaptive quantization location. The first and very fast approach – called *logarithmic* – simply quantizes the vertex locations to b bits, $b-1$ bits, $\ldots$, 1 bit and determines the smallest number of bits, where the quantized location falls inside the validity region. There is no guarantee to find the optimal quantization location. The second approach, which is called *logarithmic 2*, quantizes each coordinate independently to b bits, $b-1$ bits, $\ldots$, 1 bit. All combinations of coordinate quantizations are examined resulting in b^3 test. The third approach always finds the optimum by checking all quantization locations that fall inside the bounding box of the validity region and is therefore called *extensive*.

For comparison to the ideas of Lee et al. we also implemented an extensive sampling strategy that searches the valid quantization location with minimal coding cost. The coding cost was determined by computing the intervals used by the arithmetic coder for the different coordinates and their packages (see section 4). This approach is by far the slowest.

3.3 Entropy and Motivation

One crucial observation is that the encoding of indices with an adaptive arithmetic coder can have different costs depending on the probability of the indices. To make a more quantitative statement let us assume that the prediction correction vectors have a distribution that is close to a normal distribution. For our analysis it is useful to re-normalize the coordinates such that the bounding box of the mesh is $[0,1]^3$. Let us further suppose that the probability distributions for the different coordinates are independent and that $\sigma \ll 1$ is the standard deviation for one of the coordinates. Then the entropy H of the coordinate indices quantized uniformly to b bits is given by equation (1.80) on page 56 of [5]:

$$H(\sigma, b) = b - s(\sigma), \qquad s(\sigma) = -\log_2 \sqrt{2\pi e}\sigma, \tag{1}$$

with the base e of the natural logarithm and the number of *saved bits* $s(\sigma)$. As σ is much smaller than one, the argument of the logarithm is also smaller than one, such that the number of saved bits is larger than zero.

This means that the entropy depends on the number of quantization bits minus the number of saved bits, which only depend on the distribution. The idea of adaptive hierarchical quantization is to interpret an index resulting from a quantization to b bits with k successive zeros at the front of its binary representation as an index from a quantization to $b-k$ bits. Such an index is denoted to be *of hierarchy level* k. If all the indices could be adaptively forced onto hierarchy level k equation 1 tells us that k bits could be saved.

$b = 12$	$\#V$	TG	AA	n	uni	log	log2	ext	ent
cow	2904			1	20.6	20.4	20.1	18.3	17.6
random	4338	15.6	11.2	1	16.2	15.8	15.4	10.7	10.2
horse	19851	15.2	12.9	5	10.7	10.3	10.2	9.1	8.7
bunny	34835			6	11.6	11.4	11.1	10.0	9.5
feline	49864	14.2	13.2	3	14.2	13.7	13.2	11.0	10.6

Table 1: Comparison of geometry cost in bpv with Touma-Gotsman (TG) and Angle Analyzer (AA). Optimal package size n for extensive approach in middle column.

But not all of the indices can be forced to a higher level. Therefore is k interpreted as the number of saved *units*, where one unit is less than one bit. The adaptive quantization algorithm maximizes the saved units in three dimensions over all valid quantization locations. Nothing has to be modified in the arithmetic coder that automatically exploits the increased probabilities of indices in higher hierarchy levels.

4 Optimization of Package Size

An arithmetic coder either has to know the probability model, or it can learn it adaptively by counting for each symbol, i.e. signed index in our case, the number of appearances and use these counts to estimate the probabilities. As for geometry coding the number of different indices is very large – for example 4096 for 12 bits quantization – the explicit encoding of the distribution would be too expensive and the adaptation does not work either as the total count for each index is too small.

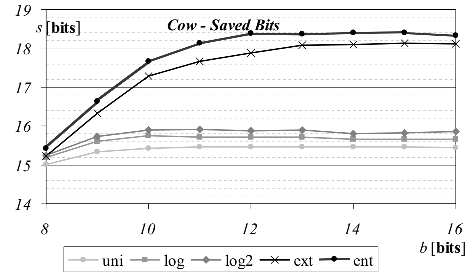
Therefore did Gumhold et al. [7] propose to group the bits of each index into packages of $n = 4$ bits, such that only sixteen symbols per package remained. For the encoding of tetrahedral mesh geometry this was the best choice for the used data sets. The same approach was investigated in this work for surface meshes. It turned out that the optimal package size n varied significantly for the different models. Therefore, we determined for each model the best package size and added this single number to the code. The package size optimization allowed to save up to two bits per vertex over a fixed package size that performs well in average.

For the proposed three approach it was possible to implement the package size optimization as an efficient post processing step. The list of signed index triples produced by the shape adaptive quantization stage was encoded arithmetically with all possible package sizes in order to determine the best choice. The overall runtime did only increase noticeably for the simple logarithmic quantization approach as this approach is already very fast.

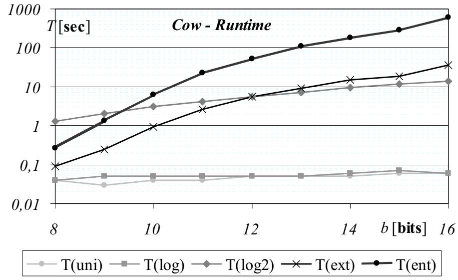
It is not possible to implement the package size optimization as a post processing step in the case of quantization based on entropy optimization. The reason is that the selection of the optimal quantization location depends on the previously seen indices and on their encoding costs. If the package size is changed, the coding cost also changes, what results in a different choice of the optimal quantization locations. In the case of the proposed hierarchical quantization scheme the cost of all quantization locations is independent of the previously encoded indices and therefore allows for further optimizations such as the proposed package size optimization.

5 Results & Conclusion

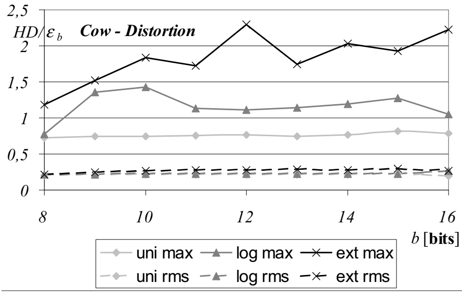
Table 1 compares the compression rates in bits per vertex of uniform quantization, logarithmic (2) hierarchical quantization, extensive hierarchical quantization and entropy optimized quantization to the Touma-Gotsman (TG) coder [14] and the Angle-Analyzer (AA) [12]. The numbers for TG and AA were taken from [12] and were not available for all the used meshes. Quantization was to $b = 12$ bits. The extensive adaptive quantization proofed significantly superior to the logarithmic approaches in all cases and slightly worse as the entropy optimization based approach. The uniform quantization approach for the horse mesh was significantly better than the original TG approach. This is probably due to the package size optimization or the use of a local coordinate system, with which the regular structure of the scanned horse and bunny meshes can better be exploited. In the middle of the table the optimal package size is tabulated. It is largest for the scanned meshes.



a)



b)



c)

Figure 2: Comparison of the different strategies (uniform sampling, logarithmic adaptive, logarithmic 2 adaptive, extensive adaptive, entropy based) in dependence of the number of quantization bits: a) saved bits per vertex s, b) runtime in seconds on logarithmic scale, c) two-sided Hausdorff-distance and RMS both divided by ϵ_b for cow mesh.

Figure 2 shows three diagrams analyzing the behavior of the proposed approaches in dependence of the number of quantization bits b. a) and b) plot the number of saved bits per vertex (compare equation 1) and the runtime of the four implemented approaches: uniform quantization, logarithmic adaptive, logarithmic 2 adaptive, extensive adaptive and entropy optimized. The uniform and logarithmic approaches show the typical behavior of most prediction based compression schemes: the number of saved bits nearly stagnates for quantization to a number of bits larger than 10. Below that all schemes seem to converge to the same rates. But the extensive adaptive approach is – similarly to the entropy optimizing approach – different for quantization to a higher number of bits b, where it achieves significant savings. Similar behavior can be found for the other models. This is very convincing and one can suspect that adaptive quantization as well as entropy optimizing quantization has a better asymptotic behavior as previous predictive coding schemes.

Figure 2 b) shows that the extensive hierarchical quantization scheme is in the order of ten times faster than the entropy optimizing approach. But the running time is still much higher as for the uniform approach. The author believes though that a more sophisticated search for the best quantization location can significantly improve the runtime of the extensive approach. Furthermore, is the decoding time not affected at all. The time for decompression of all the analyzed approaches is slightly faster than the encoding of the uniform quantization approach and reaches about 100,000 vertices per second.

A final remark has to be made on the quantization errors. Figure 2 c) plots the Hausdorff-distance divided by ϵ_b for the different approaches. This fraction should be below 1, but the adaptive approaches have larger maximum errors. This is due to the approximate computation of the validity region. The RMS as used for comparison in [12] and plotted with dashed lines was nearly the same for all approaches. One can conclude that controlling the two-sided Hausdorff-distance is much more difficult than controlling the RMS.

5.1 Conclusion and Future Work

We presented a new shape adaptive quantization method based on the approximate control of the Hausdorff-distance. It achieves compression rates similar to the direct optimization of the entropy, what is significantly slower. Furthermore, does the new method allow to optimize the resulting triples of signed integers, as the quantization process is independent of the order of vertex quantization. We exploited this property to optimize the package size used for adaptive arithmetic coding in a post processing step.

In future work the extensive search for the optimal quantization location will be speeded up with the following idea. Whenever a valid quantization position is found that allows to save k bits, no further grid locations that would save k or less bits have to be tested. The unnecessary tests can be avoided by skipping the corresponding grid locations without any additional computational cost. We believe that this significantly speeds up the extensive approach. With the entropy optimization strategy proposed by Lee et al. this acceleration strategy is not possible, as the cost of each quantization location depends on the quantization history.

Further future investigation includes a more accurate calculation of Hausdorff distance based validity. There is actually no need to construct the validity region explicitly as long as it can be safely bound. Only a validity test is necessary, where a two-sided vertex to surface distance should be sufficient for our application, as the connectivity stays the same and the vertices only move slightly.

References

[1] M. Ben-Chen and C. Gotsman. On the optimality of spectral compression of mesh geometry. *to appear in ACM Transactions on Graphics (tentative)*, 2004.

[2] M. Deering. Geometry compression. In *SIGGRAPH'95 Conference Proceedings*, pages 13–20, 1995.

[3] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH'97 Conference Proceedings*, pages 209–216, 1997.

[4] S. Gumhold. Improved cut-border machine for triangle mesh compression. In *Erlangen Workshop 99 on Vision, Modeling and Visualization*, Erlangen, Germany, November 1999.

[5] S. Gumhold, 2003. Entropy extremes for coding and visualization in 3d, Habilitation Thesis, University of Tübingen.

[6] S. Gumhold and R. Amjoun. Higher order prediction for geometry compression. In *International Conference On Shape Modelling And Applications*, pages 59–66, 2003.

[7] S. Gumhold, S. Guthe, and W. Strasser. Tetrahedral mesh compression with the cut-border machine. In *Visualization'99 Conference Proceedings*, pages 51–58, 1999.

[8] S. Gumhold and W. Strasser. Real time compression of triangle mesh connectivity. In *SIGGRAPH'98 Conference Proceedings*, pages 133–140, 1998.

[9] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *SIGGRAPH'00 Conference Proceedings*, pages 279–286, 2000.

[10] B. Kronrod and C. Gotsman. Optimized compression of triangle mesh geometry using prediction trees. In *Proceedings of International Symposium on 3D Data Processing Visualization and Transmission*, pages 602–608, June 2002.

[11] E. Lee and H. Ko. Vertex data compression for traingular meshes. In *Proceedings of Pacific Graphics*, pages 225–234, 2000.

[12] H. Lee, P. Alliez, and M. Desbrun. Angle-analyzer: A triangle-quad mesh codec. *Computer Graphics Forum*, 21(3):383–383, 2002.

[13] G. Taubin and J. Rossignac. Geometric compression through topological surgery. *ACM Transactions on Graphics*, 17(2):84–115, 1998.

[14] C. Touma and C. Gotsman. Triangle mesh compression. In *Graphics Interface'98 Conference Proceedings*, pages 26–34, 1998.