RATE-DISTORTION OPTIMIZATION IN DYNAMIC MESH COMPRESSION

K. Müller, A. Smolic, M. Kautzner and T. Wiegand

Fraunhofer Institute for Telecommunications, Heinrich-Hertz-Institut
Image Processing Department
Einsteinufer 37, 10587 Berlin, Germany
{kmueller/smolic/kautzner/wiegand}@hhi.de

Keywords: Linear Predictive Coding, Multidimensional Coding, 3D Compression, Rate-Distortion Optimization,

ABSTRACT

Recent developments in the compression of dynamic meshes or mesh sequences have shown that the statistical dependencies within a mesh sequence can be exploited well by predictive coding approaches. Coders introduced so far use experimentally determined or heuristic thresholds for tuning the algorithms. In video coding rate-distortion (RD) optimization is often used to avoid fixing of thresholds and to select a coding mode. We applied these ideas and present here an RD-optimized mesh coder. It includes different prediction modes as well as an RD cost computation that controls the mode selection across all possible spatial partitions of a mesh to find the clustering structure together with the associated prediction modes. The structure of the RD-optimized D3DMC coder is presented, followed by comparative results with mesh sequences at different resolutions.

1. INTRODUCTION

Until recently transmission of compressed 3D objects is not prevalent, since most distributed 3D applications, like massive multiplayer games or virtual reality navigation don't require real-time transmission of synthetic 3D geometry. Therefore mostly uncompressed data is transmitted. However certain new applications, such as Free Viewpoint Video [8], use synthetic 3D geometry in addition to video texture sequences and require constant real-time transmission. Further transmission of 3D graphics to mobile receivers such as mobile phones becomes more and more popular. Therefore, efficient compression of 3D geometry data becomes more and more important.

MPEG, as an ISO standardization body, already specified a standard for compression of static 3D meshes in MPEG-4 called 3D Mesh Compression (3DMC) [4]. Later, a method for compression of dynamic meshes, namely Interpolator Compression (IC), was included in an amendment called Animation Framework eXtension (AFX) [3].

These MPEG specifications are regarded as reference for compression of static and dynamic 3D meshes and perform very well among other algorithms introduced so far, which are summarized in [1]. Although AFX IC already provides good compression for dynamic meshes, we have shown in prior work that there is considerable room for improvement. We recently introduced dynamic 3DMC (D3DMC) for compression of dynamic meshes [7]. The algorithm exploits spatial correlation within a mesh much better than the standard approach by using spatial clustering algorithms [9]. Although D3DMC already gave better results on average, compared to the MPEG approach, there was further room for improvement, mainly in 3 areas:

1. A dynamic quantization allocation that is controlled by the degree of 3D vertex motion.
2. Different prediction modes in order to improve results for high-quality 3D mesh reconstruction. Here the previous fixed version of D3DMC performed slightly worse than AFX IC due to bit spending on the octree vectors as well as on the subdivision structure, whereas AFX IC only codes the difference vectors. For this, a first approach for combining spatial clustering with direct coding of differential motion vectors (direct mode) was introduced in [10].
3. Finally an advanced mode selection that decides between different spatial clustering algorithms at each hierarchy level based on a rate-distortion optimization.

By integrating these ideas, we developed an RD-optimized version of D3DMC that delivers superior coding results for all data rates. The algorithm measures the data rate for each reconstruction unit and constant reconstruction quality at this level and selects the prediction mode from 3 candidate modes:

1. **Direct Coding** of differential vectors. This mode is applied, if the motion vectors within the currently analyzed spatial volume are very different.
2. **Trilinear Interpolation** of all differential motion vectors from the 8 corners. This mode is selected, if the motion vectors exhibit a moderate smooth variation across the considered volume.

3. **Mean Replacement** of all motion vectors by their mean vector. This mode is selected, if all motion vectors within a volume are very homogeneous.

The whole process is analyzed for each single volume, starting from the fully subdivided volume, where each spatial cell only contains 8 vectors. This minimum number is required, since trilinear interpolation uses 8 corner vectors, such that this mode becomes similar to direct coding at this minimum number. Therefore, only Mean Replacement and Direct Coding need to be evaluated at this level. The analysis continues up to the single volume, where the entire object's motion vectors are contained. Finally, a subdivision scheme with the appropriate spatial clustering modes is selected for a given reconstruction quality.

2. CODING STRUCTURE

The general structure of the RD-optimized coder still applies a DPCM structure, as already introduced in [7]. For the I mesh, i.e. the first mesh within a sequence, conventional static 3DMC is used which also codes the mesh connectivity. Note, that we target temporal consistent 3D meshes with constant connectivity over time. In case of a change the algorithm needs to be reinitialized by a new I mesh. I meshes may further be introduced at any time to provide random access to the bitstream or to improve the overall performance. New elements as introduced in the last section have been integrated as shown in Fig. 1 in the block "RD-Optimization". A detailed graph of this block is shown below.

Here, the analysis-and-reconstruction block from the fixed D3DMC, where only octree clustering together with trilinear interpolation is applied, is extended to the RD optimization block. The RD optimization takes all differential vectors $d(t)$ and outputs the arithmetically coded data $y(t)$ of a current mesh. For best compression results, context adaptive arithmetic binary coding (CABAC) [6] is used, which optimally adapts to the mesh difference vector statistics. CABAC was also successfully applied to video compression and was therefore standardized for H.264/MPEG4-AVC [5]. Furthermore, the RD-optimization also outputs the estimated differential vectors $\hat{d}(t)$ for the following mesh at time t+1. As already indicated, the RD-optimization contains 3 different prediction modes, which are controlled and selected by the central "RD-Calculation" block. Since the mode selection is based on the RD decision, it requires the current distortion and rate. For that, distortion is calculated from input and predicted vectors $d(t)$ and $\hat{d}(t)$ respectively, while the actual rate is taken from the arithmetically coded data $y(t)$. Therefore, the arithmetic coder needs to be included into the RD analysis loop. With these inputs, the RD calculation block obtains the coder settings and controls the octree subdivision block for the appropriate spatial partitioning as well as the mode selection via the signal flow switches.

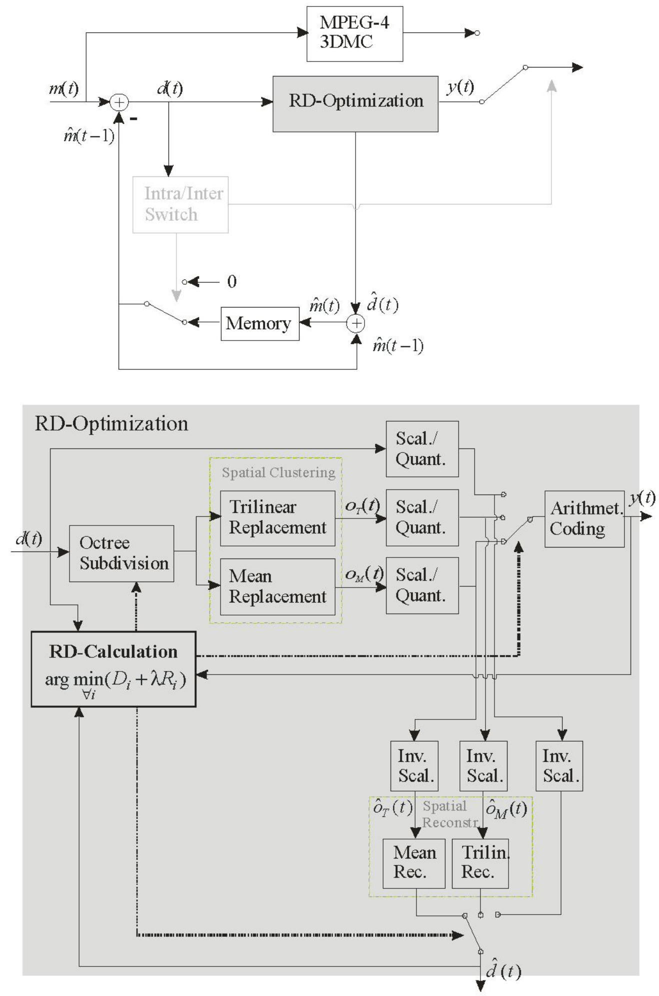

Fig. 1: Coding Structure for the RD-optimized Dynamic 3D Mesh Coder (*top*) and detailed structure of RD-optimization.

3. COMPARATIVE CODING RESULTS

For comparison of the proposed RD-optimized D3DMC, the "humanoid" test set with different resolutions was selected, which is an animated sequence of 399 frames at resolutions of 498, 1940 and 7646 vertices per mesh. The sequence is represented by 46 keyframes, which are also available for mesh coding and decoding. All other meshes are linearly interpolated during rendering from the CoordinateInterpolator syntax from VRML [2], in which the animation is provided. In the following section 3.1, the RD-based mode selection at a given overall distortion is analyzed, while section 3.2 provides comparative results with fixed D3DMC and AFX IC at different mesh resolutions.

3.1. Point Cloud Creation

As already described, the coder performs RD optimization for all possible spatial subdivisions and this way determines the combination of subdivisions together with the appropriate spatial clustering mode. One example of an obtained RD point cloud is shown in Fig. 2. For visualization purposes, the distortion axis was logarithmically scaled to better show point resolution at low distortion errors, where one of the points is selected as the minimization of RD costs at that data rate.

Here, the different distortion settings are visible by vertical point clusters with their values never falling below the appropriate local distortion values. The final value is taken from that entire point cloud as the point with minimum rate-distortion value $D_A+\lambda R$. In the example in Fig. 2, this is the single leftmost point at $D_A = 0.0005$.

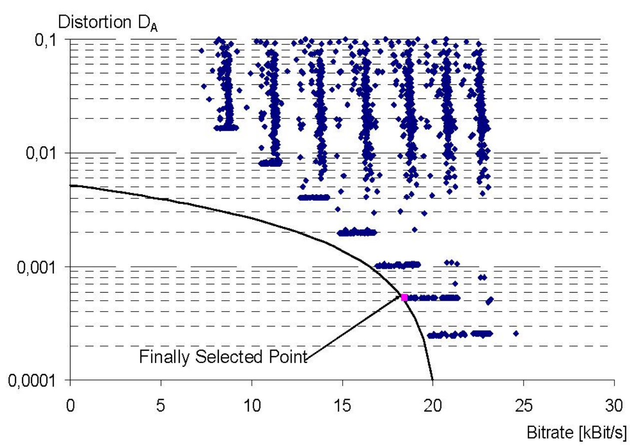

Fig. 2: RD point cloud (logarithmic scale) and finally selected rate point at a given maximum distortion (0.0006) and minimum bit rate.

3.1. Coding Results

After RD optimization, the obtained coding curves were compared against fixed D3DMC and AFX IC. The first graph in Fig. 3 shows the results for the coarsest resolution with 498 vertices. Here, the fixed D3DMC performed worse, than the standard AFX IC, since a relatively large percentage of the data rate is used for coding the spatial clustering structure. In comparison to that, the improved RD-optimized D3DMC performs similar to AFX IC at bitrates above 10 kbit/s and even better below. The improvement of the coder in comparison to the fixed version for low-resolution meshes mainly comes from the choice between different clustering modes, where a larger partition of the mesh is directly differentially coded, such that the tree structure for subdivision description is reduced.

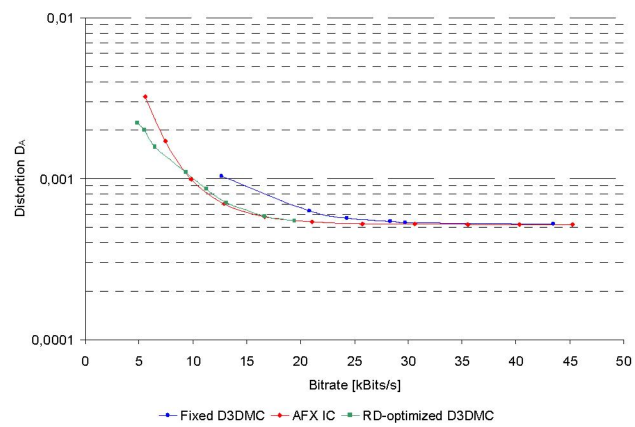

Fig. 3: Distortion D_A over bit rate for Fixed and RD-optimized D3DMC and AFX IC, L1Humanoid_L3, 46 keyframes, 498 vertices.

In the second example at medium mesh resolution of 1940 vertices, the RD-optimized coder performs again better than the fixed version. A significant improvement is obtained at higher data rates and lower distortion, where the fixed coder performed slightly worse than the standard AFX IC. Here, the RD-optimized D3DMC mainly codes the differential mesh motion vectors directly, thus omitting again a rather large and detailed subdivision structure. Both methods perform better at lower data rates, since here both D3DMC versions really benefit from the spatial clustering in combination with trilinear interpolation or mean replacement, where a whole spatial section of differential motion vectors is coded by 8, respectively 1 substitution vector(s).

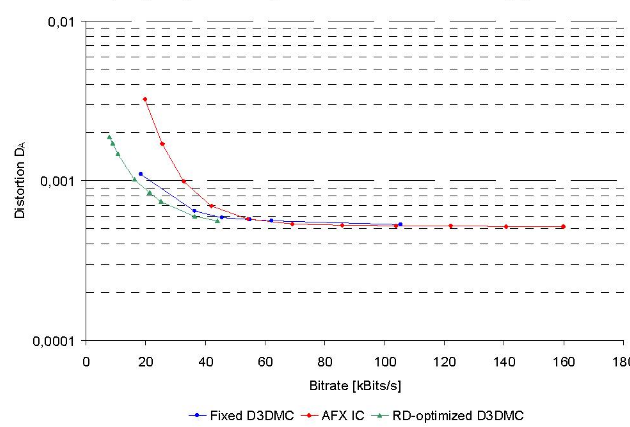

Fig. 4: Distortion D_A over bit rate for Fixed and RD-optimized D3DMC and AFX IC, L2Humanoid_L3, 46 keyframes, 1940 vertices.

In the third case for highest mesh resolution of 7646, the coding gain of both D3DMC versions against AFX IC becomes even larger, as even more motion vectors with similar statistical properties can be clustered together and coded by very few substitution vectors.

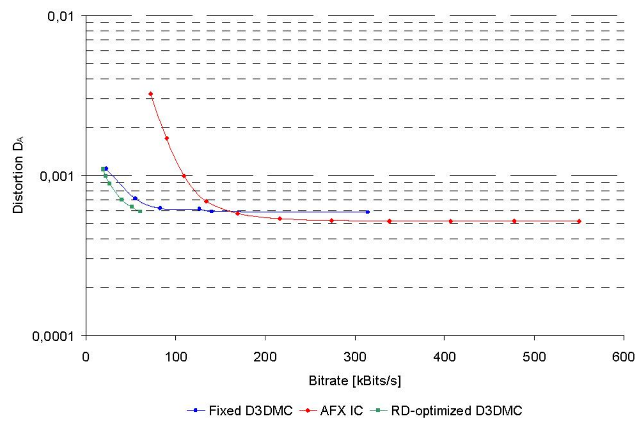

Fig. 5: Distortion D_A over bit rate for Fixed and RD-optimized D3DMC and AFX IC, L3Humanoid_L3, 46 keyframes, 7646 vertices.

Overall, the proposed optimized D3DMC approach outperforms fixed D3DMC as well as AFX IC. The main improvements against the fixed version are in the area of high-quality coding at higher data rates, where the fixed version had to allocate a rather large portion of the total bitrate for the subdivision structure. The overall better performance of the RD-optimized D3DMC comes from the combination and optimization of coding technology from both, fixed D3DMC as well as from AFX-IC. While fixed D3DMC only uses spatial clustering with trilinear interpolation, AFX IC is purely based on arithmetic coding of differential vectors. These modes are all included in the RD-optimized D3DMC, such that the improved coding results are not surprising.

4. CONCLUSIONS

In this paper we have presented an RD-optimized version of the D3DMC coder, which utilizes different prediction modes for the compression of differential motion vectors within a sequence of time-consistent meshes. The appropriate mode selection is controlled via computation of RD costs by selecting the coding mode with minimal costs including all possible spatial subdivisions of the mesh sequence, starting from the small cell mode selection up to the global bounding cube with the entire set of differential vectors.

The obtained results show that the RD-optimized D3DMC coder always performs superior in comparison to the fixed version, where spatial subdivision is controlled by a global error value, and AFX IC as the current standard for dynamic mesh compression. Although fixed D3DMC already performs better than AFX IC on average, there are still losses for very high data rates as well as for very low mesh resolutions due to the additional partitioning information that needs to be transmitted. These problems are solved by using different prediction modes and thus combining the advantages of both coders into one single design.

Future work will include further performance optimization through adding or replacing prediction modes. Moreover, since the mode selection results as found in our experiments indicate a certain pattern for high/low resolution meshes as well as high/low reconstruction quality, a pre-processing analysis about the mesh sequence characteristics could indicate, which restrictions could be imposed onto the RD-optimized D3DMC coder, in order to speed up processing.

5. ACKNOWLEDGEMENTS

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV.

We would like to thank Alexandru Salomie from Vrije Universiteit Brussel for providing the Humanoid sequence.

6. REFERENCES

[1] P. Alliez, "Recent Advances in Compression of 3D Meshes", *Proc. EUSIPCO 2005, 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[2] ISO/IEC DIS 14772-1, "The Virtual Reality Modeling Language", April 1997.

[3] ISO/IEC JTC1/SC29/WG11 "Animation Framework eXtension Core Experiments Description", Doc. N6988, Hong Kong, China, 2005.

[4] ISO/IEC JTC1/SC29/WG11, "ISO/IEC 14496-16/PDAM1", Doc. N6544, Redmond, WA, USA, 2004.

[5] ITU-T Recommendation H.264 & ISO/IEC 14496-10 MPEG4-AVC, Advanced Video Coding for Generic Audio-Visual Services, 2003.

[6] D. Marpe, H. Schwarz and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/MPEG4-AVC Video Compression Standard", *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620-636, 2003.

[7] K. Müller, A. Smolić, M. Kautzner, P. Eisert, and T. Wiegand: „Predictive Compression of Dynamic 3D Meshes", *Proc. ICIP2005, IEEE International Conference on Image Processing*, Genoa, Italy, Sept. 2005.

[8] A. Smolic, K. Müller, P. Merkle, M. Kautzner, and T. Wiegand: "3D Video Objects for Interactive Applications", *Proc. EUSIPCO 2005, 13th European Signal Processing Conference*, Antalya, Turkey, Sept. 2005.

[9] J. Zhang, C. B. Owen, "Hybrid Coding for Animated Polygonal Meshes: Combining Delta and Octree", *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume I*, pp. 68-73, 2005.

[10] J. Zhang and C. B. Owen, "Octree-based Animated Geometry Compression", *DCC'04, Data Compression Conference*, Snowbird, Utah, USA, pp. 508-517, 2004.