

Rate-distortion optimization for progressive compression of 3D mesh with color attributes

Ho Lee · Guillaume Lavoué · Florent Dupont

© Springer-Verlag 2011

Abstract We propose a new lossless progressive compression algorithm based on rate-distortion optimization for meshes with color attributes; the quantization precision of both the geometry and the color information is adapted to each intermediate mesh during the encoding/decoding process. This quantization precision can either be optimally determined with the use of a mesh distortion measure or quasi-optimally decided based on an analysis of the mesh complexity in order to reduce the calculation time. Furthermore, we propose a new metric which estimates the geometry and color importance of each vertex during the simplification in order to faithfully preserve the feature elements. Experimental results show that our method outperforms the state-of-the-art algorithm for colored meshes and competes with the most efficient algorithms for non-colored meshes.

Keywords 3D mesh · Progressive compression · Rate distortion optimization · Adaptive quantization · Color attributes

1 Introduction

3D meshes are becoming increasingly used in numerous applications, especially in the field of scientific visualization

where they constitute one of the principal representation to model outcomes from post-processing algorithms. Also, with the technological development of acquisition and computational devices, the complexity of 3D meshes has rapidly increased to represent objects with more and more details. The ever-increasing amount of these meshes, which generally include attributes such as colors or various scalar fields, requires efficient compression techniques to reduce the storage size and especially to improve the transmission time over networks, which is often the bottleneck for collaborative visualization and manipulation.

Two different approaches exist for mesh compression: single-rate approaches and progressive approaches. The benefit of the single-rate approaches is that they give a generally high compression rate. However, the reconstruction is only available when all of the data are received at the decompression stage. The progressive approaches are more relevant, especially in a scenario of remote collaborative visualization that not only requires fast data transmission but also an adaptation of the complexity of the model to the resource capacity of the client terminal. Indeed, progressive compression approaches allow achieving high compression ratio and also to produce different levels of detail. They provide the possibility to rapidly obtain a coarse version of the original object and to refine it gradually until the level of detail is the most suitable for the terminal.

The rate-distortion (R-D) trade-off is an important criterion for assessment of progressive compression methods, since the user wishes to obtain the intermediate mesh of the best quality for a given amount of received bits. Therefore, optimization of the R-D performance is an essential issue in progressive transmission techniques.

In the literature, the encoding of color data attached to mesh vertices has often been neglected in spite of its contri-

H. Lee (✉) · F. Dupont
Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205,
69622, Lyon, France
e-mail: ho.lee@liris.cnrs.fr

F. Dupont
e-mail: florent.dupont@liris.cnrs.fr

G. Lavoué
Université de Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205,
69621, Lyon, France
e-mail: guillaume.lavoue@liris.cnrs.fr

bution to the visual quality and its considerable size regarding the geometry and the connectivity.

In this context, the objective of this work is to propose a new R-D optimization method for meshes with color attributes. Our algorithm encodes the original mesh in a lossless manner, allowing only a negligible quantization error.

Our main contributions are as follows.

1. We propose an R-D optimization technique based on the adaptation of geometry quantization for meshes without color. The use of a new determination of optimal quantization based on an analysis of geometric properties has considerably reduced the calculation time compared to our previous work [19].
2. We extend our R-D optimization technique to colored meshes by adapting both the geometry quantization and the color quantization. We also propose complexity measures for the geometry and the color for prompt determination of their optimal quantization for all intermediate meshes.
3. A new metric which measures the importance of a vertex regarding the geometry and the color is also proposed to more faithfully preserve feature elements, especially at low resolutions.
4. Although our approach is based on the compression method of Alliez and Desbrun [3], all our contributions cited above are generally applicable and thus can be easily adapted to other methods.

In the next section, we provide a review of the state-of-the-art of 3D mesh compression, associate attribute data compression, and R-D optimization. In Sect. 3, we describe the base algorithm used for our approach. In Sects. 4 and 5, we present our R-D optimization techniques and results for meshes without and with color data, respectively. The conclusions are drawn in Sect. 6.

2 Related work

The first work on mesh compression was a single-rate algorithm, introduced by Deering [9] based on generalized triangle strips. Then, Taubin and Rossignac [29] proposed a topological surgery approach by encoding two spanning trees. Rossignac introduced the *EdgeBreaker* algorithm [26], which encodes the mesh connectivity by storing five symbols obtained by using an edge conquest. This method guarantees 3.67 bits per vertex (bpv) as a worst case for the connectivity coding. Gumhold and Straßer introduced a similar algorithm called *Cut-border machine* [11].

The valence-driven algorithm proposed by Touma and Gotsman [30] is considered to be one of the most efficient single-rate algorithms in terms of compression rates. Based

on the edge-centered conquest, they encode the connectivity by storing valence codes of mesh vertices with some additional incident codes. Therefore, this algorithm is mainly efficient for regular meshes. In [4], an improved valence-driven algorithm is presented by Alliez and Desbrun.

All the algorithms explained above are single-rate lossless algorithms which tolerate only quantization error of the mesh geometry. Isenburg et al. [14] proposed a method to encode, in a lossless way, vertices coordinates, for the purpose of preserving the original float precision.

For progressive encoders, Hoppe introduced the first algorithm in [13], called *progressive mesh*. A given mesh is successively simplified by applying a sequence of edge collapses which removes one vertex and two faces adjacent to the edge. The reconstruction, at the decompression stage, is accomplished by the vertex split operation. To optimize the approximation quality of intermediate meshes, each edge collapse is chosen based on geometric criteria at the expense of some overhead of coding rate due to the cost needed for the localization of the inverse operations.

Several approaches were then proposed to improve the compression ratio. Taubin et al. [28] introduced a progressive algorithm which uses the forest split operation, reducing the connectivity coding cost to 7–10 bpv. In [22], Pajarola and Rossignac grouped vertex splits into batches by marking vertices to be split using 1 bit at each resolution, performing 7 bpv for the connectivity. The position of a new vertex is coded based on the butterfly-like prediction using its neighboring vertices. Karni et al. [15] proposed a similar approach: a sequence of edges which traverses all mesh vertices is first built. Then a group of edge collapses is performed to every pair of adjacent vertices to generate different resolutions. Due to the good locality and continuity properties of this sequence, applying edge collapses within the sequence leads to an improvement of compression rates and rendering speed.

In [8], Cohen-Or et al. proposed the patch coloring technique for the progressive transmission. This algorithm successively removes a set of vertices by using vertex removal operations. The hole left by the deletion is retriangulated in a deterministic way. The coding cost of the connectivity is achieved to 6 bpv. Alliez and Desbrun [3] proposed a valence-driven progressive coder which benefits from the native compact distribution of vertex valences to efficiently encode the connectivity. This algorithm is also based on vertex removal, and a deterministic retriangulation tries to maintain the mesh regularity as much as possible to optimize the approximation quality of intermediate meshes and the coding cost of the connectivity. The connectivity is compressed to an average of 3.7 bpv. Recently, Ahn et al. [2] improved the coding rates and rate-distortion trade-off of [3] by using a connectivity-based Karhunen-Loeve transform and a bit plane coder for the geometry encoding.

All the progressive algorithms described above are connectivity-driven techniques, meaning that priority is given to the connectivity coding. However, as geometry data occupy usually more space than connectivity data in the compressed file, geometry-driven algorithms have been introduced more recently.

Gandoin and Devillers [10] proposed the first geometry-driven algorithm based on the kd-tree space subdivision. This algorithm is performed in two passes; the first pass consists of encoding only geometry data. They recursively divide the space in two cells until there remains only one vertex in each cell. The number of vertices in one cell is then encoded. The second pass encodes the connectivity changes caused by each cell subdivision using a generalized vertex split. In terms of compression ratio, this technique outperforms connectivity-driven algorithms and can even compete with the single-rate coders like [11][30][26]. Peng and Kuo [25] introduced an improved geometry-driven approach based on the octree subdivision. Efficient prediction for both the connectivity and the geometry brought an improvement of 10 to 20 % compared to [10]. Geometry-driven algorithms generally outperform connectivity-driven algorithms in terms of lossless compression ratio. However, their efficiency is limited for progressive compression since they provide quite poor approximation quality for intermediate meshes.

More recently, Valette et al. [31] proposed a progressive approach based on a reconstruction scheme. They start from a coarse version of the original model which is refined progressively by inserting a vertex to the longest edge using edge split operation, aiming to generate uniformly sampled intermediate meshes. By restoring the original connectivity only at the end of the transmission and by adapting the quantization level by transmitting progressively vertex coordinates, this algorithm performs efficiently in terms of rate-distortion trade-off. In [32], an other technique which transmits gradually vertex coordinates is proposed. Nevertheless, this technique is not fully progressive as the connectivity is kept unchanged during the transmission.

Peng et al. [24] proposed a new progressive technique which generates different levels of detail by successively applying a vertex set split using a generalized Lloyd algorithm. In this technique, they took into account geometric features to optimize the quality of each level of detail and to perform adaptive quantization. This feature-driven optimization of model representation in the whole hierarchy together with efficient prediction of the geometry and the connectivity achieves outstanding rate-distortion performance.

When the user needs to preserve only the *shape* of the 3D model, a remeshing algorithm can be performed prior to the compression step to change the mesh connectivity to a structured connectivity. The use of transformations like wavelet is then applicable, permitting one to considerably reduce the

coding rate. The first progressive algorithm based on the wavelet transform was proposed by Khodakovsky et al. [17]. Later on, Khodakovsky and Guskov [16] introduced a more efficient wavelet-based coder using the normal mesh representation [12]. Recently, Valette and Prost [33] extended the wavelet-based algorithm to handle irregular meshes. Mamou et al. [21] used a shape approximation of the original mesh to encode the mesh geometry. Their method gives a satisfactory result; however, its computational complexity is high, and only the geometry is refined during the reconstruction.

Up to the present, only a few researchers focused on coding of color information. Among the single-rate compression techniques, only [9] [5] [29] proposed encoding methods for vertex-bind color information. However, the prediction used for the color encoding is the same as for the geometry encoding regardless of its different nature. Recently, Ahn et al. [1] and Yoon et al. [34] introduced new methods adapted for color data. Ahn et al. [1] used a mapping table method based on the vertex layer traversal algorithm. Instead of encoding color coordinates of each vertex, they encode the index of the vertex color in the mapping table. A color value in the mapping table is encoded when it appears for the first time during the traversal. Yoon et al. [34] introduced a prediction method using connectivity and geometry information of neighboring vertices. They consider different weights for the neighboring vertices based on the angle analysis. Then, the color value of the current vertex is predicted from weighted averaged color values. For the progressive algorithms, Cirio et al. [7] recently introduced a technique that handles color information. However, the quality of intermediate meshes is quite poor since their algorithm is based on a geometry-driven algorithm [10].

Bit allocation is a technique which allows optimizing the rate-distortion trade-off. King and Rossignac [18] introduced an approach which determines the optimal relationship between the number of vertices and the number of bits of vertex coordinates optimizing the distortion for a given bit rate, based on a shape complexity measure. For the wavelet-based compression, Payan and Antonini [23] proposed an optimized bit allocation. The best quantizer for wavelet coefficients of each sub-band is determined so as to minimize the geometric error for a given bit rate. However, these two bit allocation techniques are not adapted for the progressive transmission since even if more bits are received the refinement is not possible.

3 Base algorithm

Generally, connectivity-guided algorithms produce better approximation quality for the intermediate meshes than geometry-guided ones, especially at low resolutions. For this reason, our work is based on the algorithm of Alliez and

Desbrun [3], which is quite simple and nevertheless one of the most efficient connectivity-driven algorithm. This algorithm extends the concept of the valence-driven single-rate approaches [30] [4] for progressive encoding. The valence-driven scheme profits from the beneficial statistical distribution of vertex valences to efficiently encode the connectivity information.

The algorithm of Alliez and Desbrun [3] successively applies two conquests which remove a set of vertices, generating different levels of detail (LODs): decimation and cleansing conquests. The decimation conquest traverses the mesh in a deterministic way using a gate-based traversal; when the valence code of the current front vertex is inferior or equal to 6, this vertex is removed and then the hole left is retriangulated. Similarly, the cleansing conquest removes vertices of valence equal to 3.

For regular meshes, the combination of these two conquests performs the inverse $\sqrt{3}$ subdivision as shown in Fig. 1, and the connectivity coding is also optimal for this case. For non-regular meshes, the retriangulation follows a deterministic rule so that the mesh connectivity is kept as regular as possible during the simplification process. For these meshes, besides valence codes of the removed vertices, incident codes need to be encoded, introducing some coding overhead.

For the geometry coding, the authors first applied a global and uniform quantization to the mesh vertex coordinates. Assuming the smoothness and the regularity of the mesh, the authors use a local prediction which is a barycentric prediction. When a vertex is removed, its position is predicted from the average position of its 1-ring neighboring vertices and the difference is encoded. To further reduce the bit rate, they also separate tangential and normal components as pointed in [17], by adopting a local frame (Frenet frame). In our previous work [19], we proposed an improved geometric coder based on a discrete bijection which brings an overall coding gain between 3 and 20%. In this paper, we use the mesh traversal and the connectivity encoding method of [3] and the geometry coder of [19].

The principal objective of our work is to propose an efficient progressive encoding method for colored meshes. Nevertheless, as most researchers focus only on non-colored

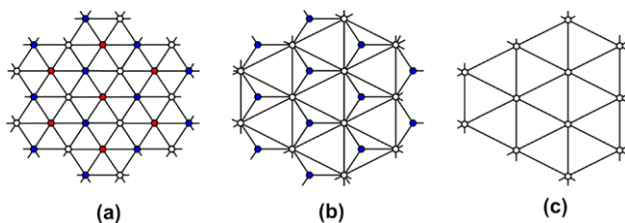


Fig. 1 A regular mesh (a) is simplified by the decimation conquest (b) and the cleansing conquest (c) in the algorithm of [3]. We can see that the resulting mesh is also regular

meshes, we have also derived a simpler version for meshes without attributes, which is presented in the next section.

4 Rate-distortion optimization for meshes without color attributes

In this section, we focus on the optimization of the rate-distortion (R-D) trade-off for meshes without color data.

One important factor of the R-D performance is the quantization precision of the mesh vertex coordinates which is defined by the quantization step applied at the beginning of the algorithm. In our approach, the quantization precision is adapted to each LOD according to its number of elements to optimize the R-D performance.

The high precision induced by the initial quantization is indeed necessary for intermediate meshes at high resolutions where the number of mesh elements is important; however, this high precision is not needed for meshes at low resolutions. In Fig. 2, we reduce the quantization precision from 12 bits to 6 bits for the Rabbit model. This reduction of the quantization precision produces highly visible artifacts when applied on the original high resolution model (top row); however, for the low resolution model (bottom row), both geometric and visual distortions are quite similar for the 12-bits and 6-bits quantized versions.

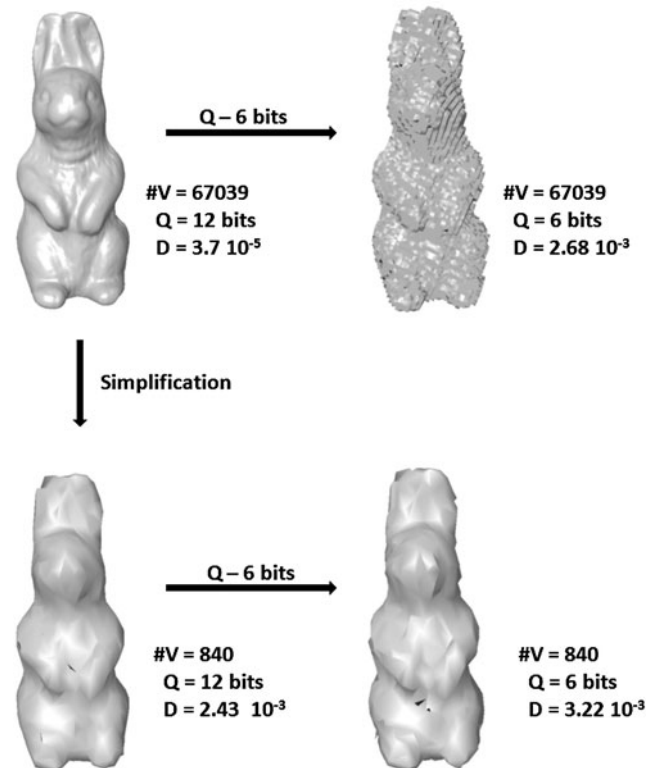


Fig. 2 Comparison of distortion values of the Rabbit model at different resolutions and with different quantization precisions

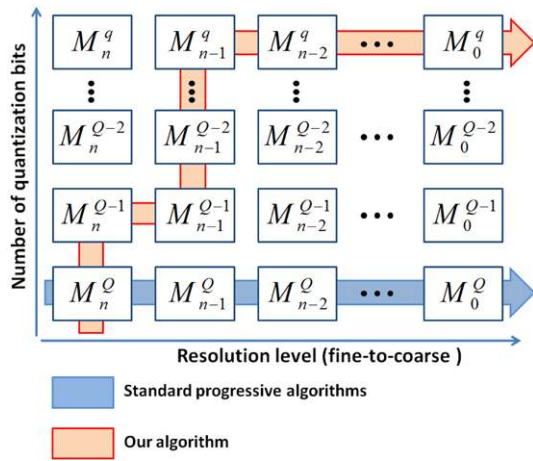


Fig. 3 Our algorithm (red arrow) also reduces the geometry quantization precision, contrary to the classic progressive algorithms (blue arrow)

This observation proves that each intermediate mesh can be adaptively quantized regarding its complexity and its number of elements, without significantly degrading the geometry quality. As the quantization is directly related to the coding rate, this adaptation of quantization will optimize the R-D performance.

Figure 3 describes the principle of our algorithm compared to classic connectivity-driven algorithms. In traditional progressive frameworks, the initial mesh, M_n^Q , which is quantized with Q bits, is iteratively simplified. After n iterations, the base mesh M_0^Q is obtained, and its quantization precision is still Q . In our work, the geometry quantization precision, Q , is also gradually reduced, allowing R-D performance optimization.

To achieve this optimization, two principal issues have to be addressed.

1. The quantization precision has to be adapted to each intermediate mesh. Hence, during the encoding process, the choice of the next operation between decimation and decrease of quantization precision has to be optimally determined at each iteration.
2. The decrease of quantization precision must also be efficiently encoded so as to avoid a significant coding overhead.

4.1 Encoding of quantization precision change

We describe here our method to decrease the quantization resolution and to efficiently encode the inverse operation.

The initial quantization using Q bits consists in dividing the bounding box of the input mesh into $2^Q \times 2^Q \times 2^Q$ cubic cells and in moving each vertex to the center of its corresponding cell. If Q is reduced to $Q - 1$, the dimension of

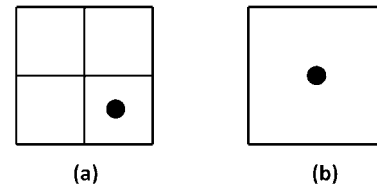


Fig. 4 At the encoding process, child cells (a) are merged to form a parent cell (b). The vertex is then moved to the center, and the correct child cell index is encoded

each cell becomes twice as long along the three axes, and each vertex is shifted to the center of the new bigger cell.

The decrease of quantization resolution can then be considered as a merging process in an octree structure as shown in Fig. 4; eight child cells (initial small cubes) are merged to form a parent cell (new big cubes). At the decoding process, inverse operations are performed to increase the quantization resolution, and the decoder needs to know for each vertex its original child cell location. For that, we encode the index of the child cell at the encoding stage.

Without any prediction, the encoding of a correct child cell index among 8 possibilities costs 3 bits. To reduce this coding cost, we adopt the prediction method from the algorithm of Peng and Kuo [25]. Based on the location and the distance of neighboring vertices, a priority value is given to each child cell. Then, the indexes of the child cells are re-ordered with respect to the priority values. The new index corresponding to the correct child cell is then encoded.

4.2 Optimal determination of the next operation

As shown in Fig. 3, at each iteration, we have to determine the next operation between decimation and decrease of quantization resolution. This choice of the next operation is a fundamental key point in our R-D trade-off optimization technique. To automatically and optimally determine the next operation, we calculate the difference of the geometry distortion ΔD compared to the original mesh M_n^Q , and the amount of bits ΔB needed to encode the inverse operation for both cases. For the decimation case, we obtain ΔB_{deci} by calculating the entropy of both the connectivity symbols and the correcting vector for all removed vertices during the two conquests. The geometric distortion ΔD_{deci} is measured as the geometric distance compared to the original mesh. Similarly, ΔB_{quan} and ΔD_{quan} are obtained for the decrease of the quantization resolution. Then, the ratios $\frac{\Delta D_{\text{deci}}}{\Delta B_{\text{deci}}}$ and $\frac{\Delta D_{\text{quan}}}{\Delta B_{\text{quan}}}$ are compared to determine the next operation. More precisely, we select the smaller one since it optimizes locally the R-D trade-off.

This automatic selection of the next operation is described in Fig. 5. In this example, decimation is chosen since $\frac{\Delta D_{\text{deci}}}{\Delta B_{\text{deci}}}$ is inferior to $\frac{\Delta D_{\text{quan}}}{\Delta B_{\text{quan}}}$.

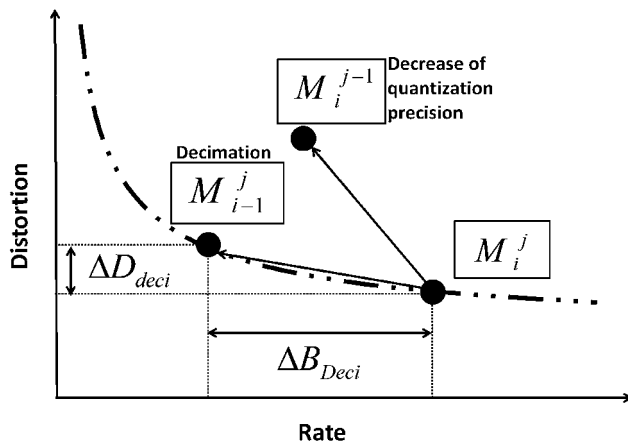


Fig. 5 Choice of the next operation between the decimation and the decrease of quantization precision

Note that, in our optimal determination scheme, root mean square (RMS) distance is used as the geometric distance obtained by the METRO tool [6]. However, other distances such as Hausdorff or any user-defined metric can also be employed.

4.3 Fast quasi-optimal determination of the next operation

Our method to optimally determine the next operation possesses one inconvenience which limits its effectiveness: the high computational time. In fact, at each iteration, our optimal determination technique needs to calculate the geometric distortion of models generated respectively by the decimation and the decrease of quantization resolution. To reduce the calculation time, we propose to determine the next operation based on an analysis of geometric complexity and on a learning process to eliminate the need for the time-consuming distortion calculation.

For the optimal case, the next operations are determined locally, at each iteration. However, we can observe using our optimal determination method that the correct quantization precision for a given level of detail (LOD) can be determined globally, without considering prior operations.

Indeed, in Fig. 6, we show three different paths obtained by our optimal method when starting from three different initial numbers of quantization bits for the Bimba model; the red, blue, and green colored lines represent, respectively, paths obtained using 12 bits, 9 bits, and 6 bits as initial quantization. The horizontal axis shows resolution levels with n as the finest level, and the vertical axis describes the quantization precision. For the resolution level n , the optimal number of quantization bits is determined globally to be 9; indeed higher initial quantization resolutions induce decreases of quantization precision, as shown by the red path. For the resolution level $n-4$, the best quantization precision is globally determined to be 6; applying initial quantization

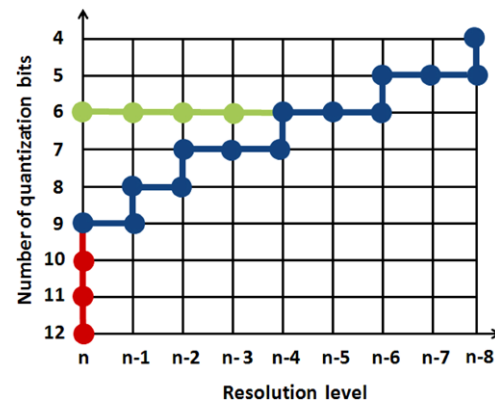


Fig. 6 Paths obtained using different initial quantizations on the Bimba

using 6 bits causes a series of decimation operations until the blue path is reached.

These observations tested on several models prove that a unique optimal quantization precision exists for each intermediate mesh independently of prior operations, and thus it is possible to determine the quantization precision based on an analysis of geometric properties.

First, we consider the number of mesh elements to estimate the appropriate quantization precision, since when the number of elements increases, the quantization has to be more precise to avoid harmful aliasing problems. We also take into account the level of complexity, since a more precise quantization is needed when the shape of a mesh is more complex.

Hence, our geometry complexity measure K_G uses the volume of the bounding box, the surface area, and the number of vertices to estimate properly the required quantization precision:

$$K_G = \frac{\text{volume of bounding box}}{\text{surface area} \otimes \text{number of vertices}} \quad (1)$$

The *surface area* is obtained as the sum of the triangles' areas. For consistency, the size of the input model is scaled so that the longest dimension of its bounding box is equal to 10. In Fig. 7, we plot values of the geometry complexity measure K_G and the corresponding numbers of quantization bits for all intermediate meshes, computed using our optimal method on 8 standard meshes.

In total, 81 points are acquired and these points constitute the data set for a learning process which establishes the appropriate relationship between the geometry complexity measure K_G and the quantization precision. Among the various types of fitting functions, the logarithm gives good results, as shown in Fig. 7. The parameters of this fitting curve are obtained using the least square method. Now a near optimal quantization precision can be determined by using

$$q_G = \text{round}(-1.248 * \log(K_G) - 0.954) \quad (2)$$

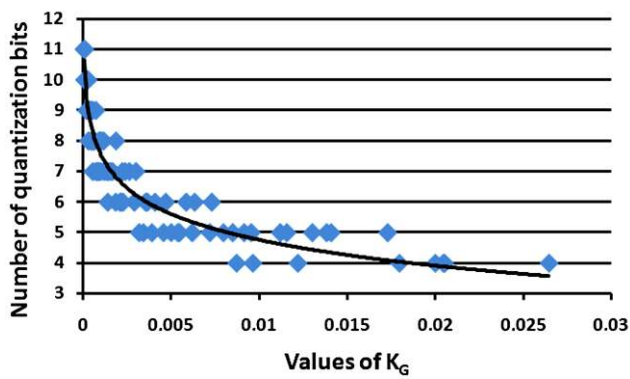


Fig. 7 Values of K_G and corresponding optimal quantization precisions obtained by using our optimal determination method on 8 reference meshes. The fitting curve to these points is also illustrated

where *round* denotes the rounding process to the nearest integer.

During the encoding process, our quasi-optimal determination method calculates the estimated quantization precision q_G at each iteration. If the current number of quantization bits Q_G is superior to q_G , a decrease of the quantization precision is performed as the next operation. Otherwise, the decimation is chosen.

The use of quasi-optimal determination of the next operation makes our R-D optimization algorithm more practical by significantly reducing the calculation time (see experimental section).

4.4 Experimental results on meshes without color attributes

In this section, we present our results compared with the algorithm of Alliez and Desbrun (AD 2001) [3], the wavemesh of Valette and Prost (VP 2004) [33], the octree-based algorithm of Peng and Kuo (PK 2005) [25], the IPR algorithm of Valette et al. (VCP 2009) [31], the method of Mamou et al. (MDCG 2010) [21], and the method of Peng et al. (PHKEG 2010) [24].

Table 1 shows the running times of our algorithm associated with optimal and quasi-optimal determination, in comparison with the base algorithm (AD 2001) [3] on a 2.80 GHz dual-core Intel CPU laptop computer with 4 GB. The timings of AD 2001 are estimated using our own implementation, which yields similar bit rates. Our algorithm using the optimal determination needs a very long time because of the distortion calculations. For instance, our optimal algorithm takes 3737.5 seconds for the Neptune, while AD 2001 requires only 4.3 seconds. However, our quasi-optimal algorithm reduces the timings significantly, consuming only 7.5 seconds for the same model. On average, our quasi-optimal algorithm is only two times slower, compared to AD 2001.

Figures 8 and 9 show the R-D curves for the Horse and the Rabbit, respectively. The distortion is measured as

Table 1 Comparison of running times between our approach and the algorithm of Alliez and Desbrun (AD 2001) [3] in seconds

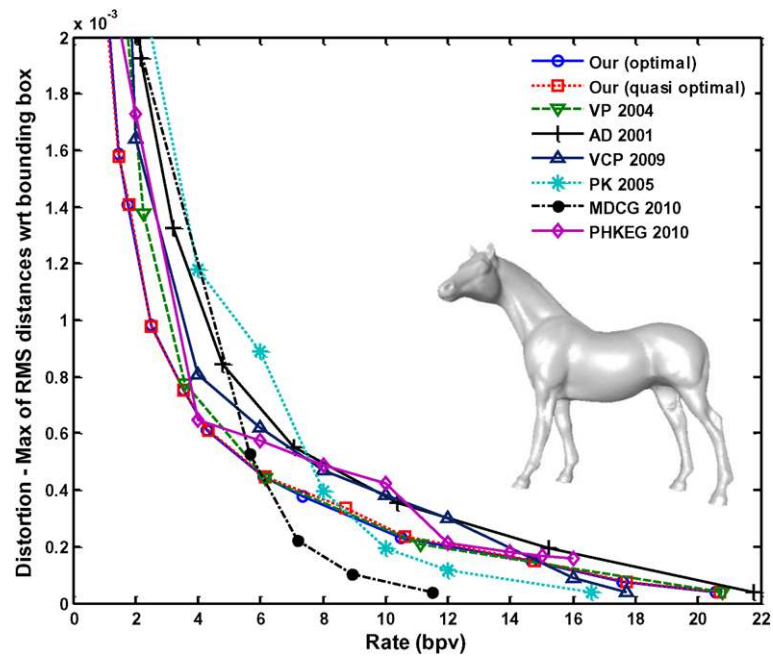
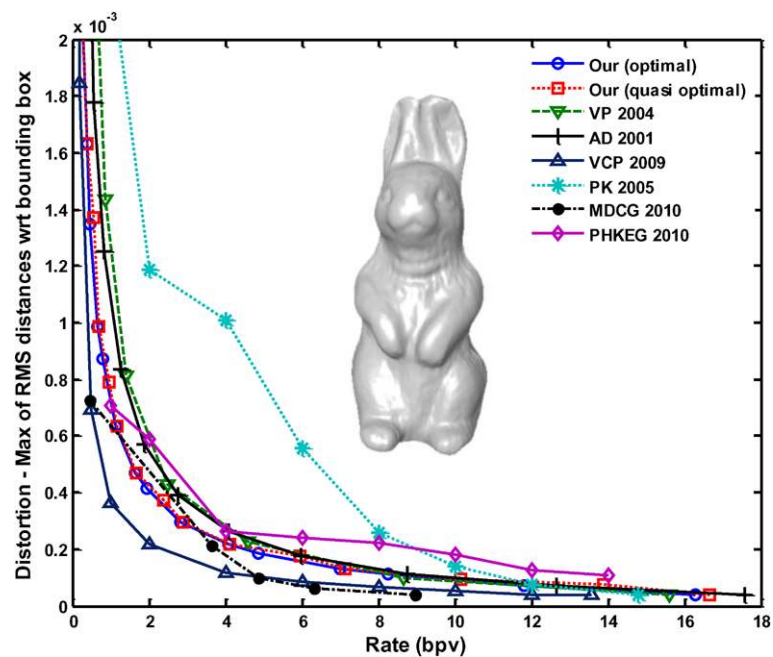
Model	#V	Q	AD 2001 [3]	Our	
				Optimal	Quasi-optimal
Fandisk	6475	10	0.1	38.8	0.2
Venusbody	11362	12	0.2	49.6	0.4
Horse	19851	12	0.5	131.8	0.8
Torus	36450	12	0.3	177.2	0.7
Rabbit	67039	12	1.0	662.1	2.4
Neptune	249425	12	4.3	3737.5	7.5

the maximum of two RMS distances (forward and backward) with respect to the bounding box using the METRO tool [6]. These meshes are quantized using 12 bits. First, we can see that our quasi-optimal algorithm yields very similar results to those of our optimal algorithm in terms of R-D performance. Note that the Horse model does not belong to the data set used to establish the quasi-optimal rule (2).

Compared to the most recent progressive algorithms, our approach produces a competitive performance. For the Rabbit model, our algorithm outperforms AD 2001, VP 2004, PK 2005, and PHKEG 2010. MDCG 2010 produces a similar result to our algorithm up to 4 bpv and yields a better result at higher rates. For this mesh, the best R-D performance is obtained by VCP 2009. For the Horse model, our algorithm obtains the best result at low rates. Above 6 bpv, MDCG 2010 produces the best result. Note that MDCG 2010 gives very impressive results because it relies on a complex shape approximation for the geometry encoding. However it requires a high computational time. Our quasi-algorithm needs less than 1 second to encode a mesh with 20,000 vertices while MDCG 2010 requires 3 minutes.

We also evaluate the performance of our optimal and quasi-optimal algorithms on the Bimba (8,857 vertices) and the David Head (24,085 vertices) models which have surfaces with more small details. Their R-D curves are illustrated in Figs. 10 and 11. For both models, our optimal algorithm significantly outperforms AD 2001 and VP 2004. Our quasi-optimal algorithm performs very similarly to our optimal one for the Bimba model, but its efficiency is degraded for the David Head model, since the parameters used to estimate the optimal quantization precision are calculated using smooth objects. However, the performance of our quasi-optimal method is relatively close to our optimal one, and it outperforms AD 2001 and VP 2004.

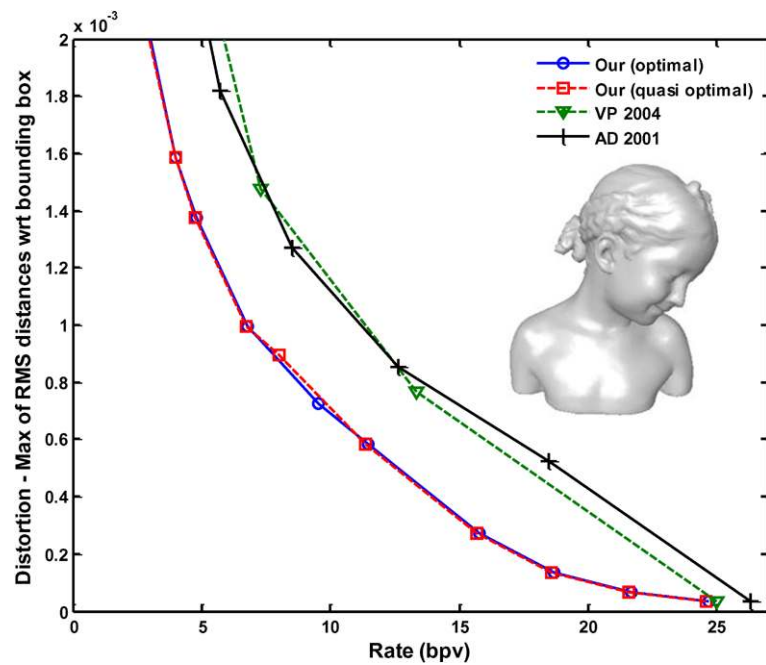
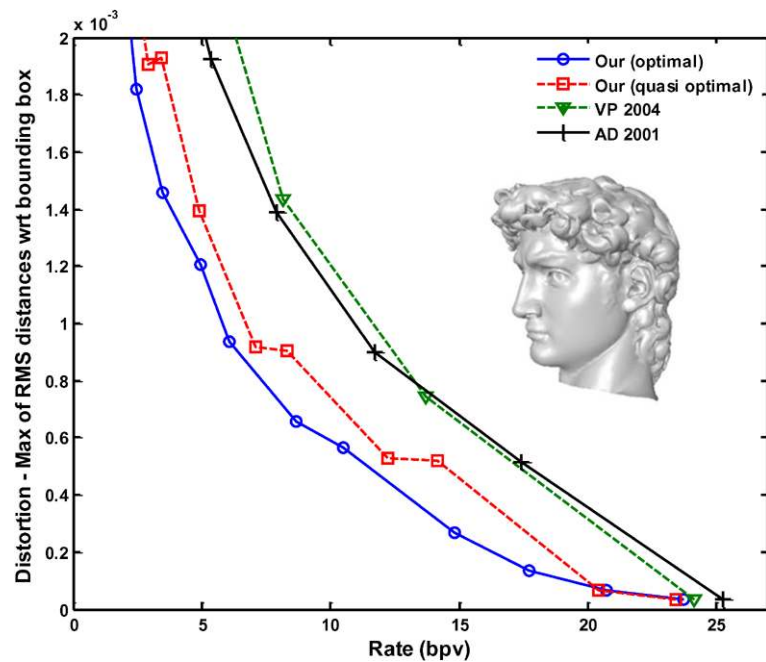
We evaluate the stability and the robustness of our quasi-optimal algorithm by varying the parameters of (2). First, we modify these parameters by multiplying them by a factor. The Venusbody and the Dinosaur models are used for

Fig. 8 R-D curve for the Horse**Fig. 9** R-D curve for the Rabbit

this evaluation. When a variation of 5% is applied, the performance of our quasi-optimal algorithm is quite identical to our optimal one. The variation of 10% affects slightly the overall R-D performance and the final compression rates. In the worst case, it leads to a local increase of approximately 3 bpv for the same level of distortion. Second, we recalculate these parameters using a new data set (five other models) for our learning process. The parameters obtained from this new data set are almost identical to the initial ones (variation is inferior to 5% for

both parameters), and the R-D performance of our quasi-optimal algorithm with these parameters is quasi similar to our optimal algorithm. These observations prove that our quasi-optimal algorithm is robust to variation of its parameters.

In Fig. 12, we show intermediate results of the Horse model for similar bit rates of our algorithm, AD 2001, and VCP 2009. At 1 bpv and about 4 bpv, we can see that our algorithm reconstructs more details (see the ears of the horse) than the other algorithms.

Fig. 10 R-D curve for the Bimba**Fig. 11** R-D curve for the David Head

5 Rate-distortion optimization for meshes with color attributes

In this section, we present our R-D optimization algorithm for meshes with color attributes. The quantization precision of color is also adapted to each intermediate mesh, improving the R-D performance.

5.1 Coding of color data

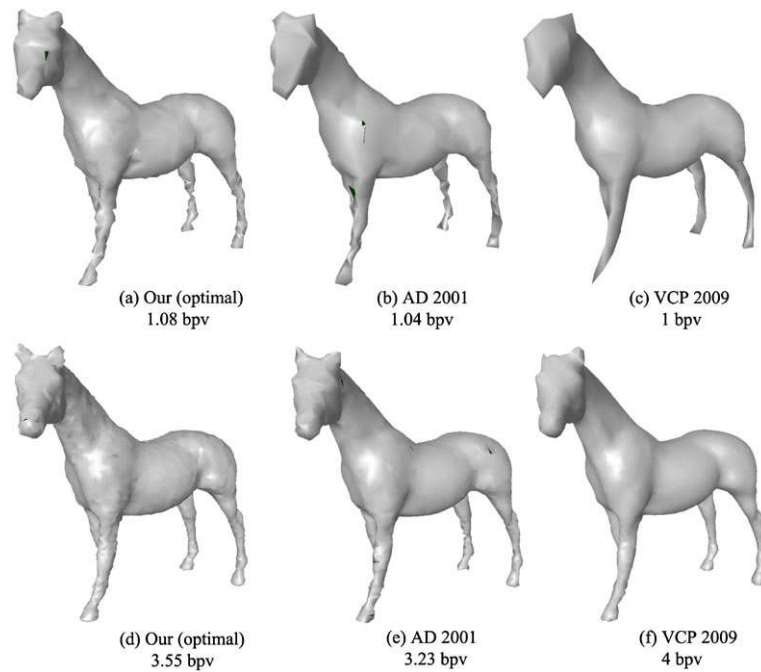
The size of color data can be as large as or even larger than connectivity and geometry without an adaptive compression

method. Therefore, we have proposed a technique to efficiently reduce these data in a previous work [20] that we describe here briefly.

At the beginning of the encoding, all colors expressed in the RGB space are transformed into the L.a.b. space. This representation is more decorrelated than the RGB space; thus it is more appropriate for data compression. After the transformation, each component L , a , and b is represented using 8 bits as in the initial RGB space.

We have observed that the color value of a vertex is generally very close to at least one of its neighboring vertices'

Fig. 12 Comparison of intermediate meshes for similar bit rates between our algorithm, the algorithm of Alliez and Desbrun (AD 2001) [3], and the algorithm of Valette et al. (VCP 2009) [31]



color. Based on this observation, we have proposed a method which predicts the color of each removed vertex by selecting the appropriate color among its neighboring vertices' color. For this selection, the average values of color components, L_{mean} , a_{mean} , b_{mean} are first calculated. Then, for each component, we select the one which is the closest to the corresponding average component among the neighboring vertices' colors. The difference between the original and the selected color is then encoded, allowing the decoder to reconstruct the color value exactly. During the decompression process, the geometry and the connectivity are first restored and the color data is added to the newly inserted vertex, allowing the progressive reconstruction.

5.2 Color metric for data-driven progressive coding

With the geometry, the color data contribute highly to the visual quality of the model. Thus, preserving the feature elements of color is an important issue for the progressive transmission to increase the approximation quality, especially at low resolution meshes.

As our connectivity encoding algorithm uses vertex removal as the simplification operator, we propose a data-driven progressive method by preventing deletion of vertices which can eventually cause a serious distortion. For that, we estimate the error caused by the removal of a vertex; then if the error is larger than a threshold, we forbid removal of this vertex.

Our metric is composed of the color error term E_{color} , and the geometry error term E_{geometry} .

$$E_{\text{total}} = E_{\text{color}} \otimes E_{\text{geometry}} \quad (3)$$

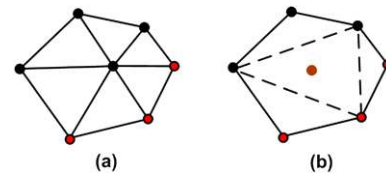


Fig. 13 E_{color} is obtained as the difference between the color of the vertex to be removed (black) and the color of the centered position of the middle triangle (brown), which is obtained by the interpolation

In Fig. 13, an example of the E_{color} calculation is described. During the encoding process, triangles adjacent to the vertex to be removed constitute the initial patch and after the removal the retriangulation leads to the final patch. We suppose first that the vertex to be removed is located near the barycenter of the initial patch. We then estimate E_{color} as the difference between the current vertex color (black) and the color of the center position of the final patch (brown), which is obtained by interpolating the vertices' colors of the middle triangle.

When we prevent a vertex removal, this penalizes the coding rate since an additional incident code is needed for the decoder. Indeed, the use of only E_{color} preserves most of the color details for all levels of details, causing a significant increase of the coding rate. Therefore, the preservation level of color details has to be adapted to the complexity of intermediate meshes.

For this purpose, we combine E_{color} with a geometry error term E_{geometry} . E_{geometry} is defined as the ratio between the average area of the triangles of the initial patch and the average area of all triangles of the current intermediate mesh. This ratio gives the relative size of the surface

affected by the vertex removal and permits us to adapt levels of color details to the resolution of intermediate meshes. Even if the color error is important, the vertex removal can be performed if the patch area is relatively small; inversely the removal of a vertex associated with a small color error is prevented if the patch area is relatively important.

The choice of the threshold differs from input meshes due to the different distributions of colors on mesh surface. To avoid user intervention to find the optimal threshold, we divide E_{color} by the average of color differences of the current intermediate mesh. The average of color difference is the average of color difference between the two vertices of all mesh edges. With this normalization, the threshold can be fixed to 0.5 for all colored models.

5.3 Encoding of decrease of color quantization precision

In [20], the color components L, a, b are quantized using 8 bits at the beginning of the encoding algorithm. As for the geometry, this high color precision is not required for intermediate meshes at low resolutions.

To decrease one bit of the color quantization precision, we consider it as a merging process in an octree structure, similarly to the geometry (Sect. 4.1). During the encoding process, each group of eight child cells is merged to form one parent cell, and the correct child cell index is encoded for each vertex.

We can observe that many vertices have the same color, unlike for the geometry, and generally there are only a few dominant colors. Since the vertices with the same color before the decrease of quantization precision have the same child cell index, instead of using one common encoder for all colors, we use one encoder for each color to reduce the coding rate.

We first traverse the mesh obtained after the decrease. Then we count the number of present colors and we attribute an index to each vertex following its color value. When a new vertex with the color index i is encountered, we use the encoder i to code its child cell index. No prediction method is employed here, but using multiple encoders allows an efficient encoding.

The reconstruction technique is similar. When an increase of color quantization is required, the number of present colors is first calculated. Then, the traversing of mesh vertices is performed by restoring their child cell index using a decoder corresponding to the encountered vertex color.

5.4 Optimal determination of the next operation

The optimal determination of the next operation for a colored mesh is also similar to the one for meshes without

color attributes. The main difference is that the next operation must be chosen from three choices: decimation, decrease of geometry quantization precision, and decrease of color quantization precision.

We aim to find the best path which optimizes the R-D trade-off by combining these three operations. To determine the best next operation, we have to compare the ratio between the bit rate and the distortion for each case, $\frac{\Delta D_{\text{deci}}}{\Delta B_{\text{deci}}}$, $\frac{\Delta D_{\text{geo}}}{\Delta B_{\text{geo}}}$, $\frac{\Delta D_{\text{color}}}{\Delta B_{\text{color}}}$, and the one with the smallest ratio value is chosen as the next operation.

To measure the distortion, we use the metric of Roy et al. [27], which measures the difference between each vertex color of one mesh and the color of the nearest point on the surface of the other mesh. This metric takes not only color differences into account but also geometry differences by the nearest point computation. Hence this metric reflects quite correctly the visual difference between two colored meshes.

5.5 Fast quasi-optimal determination of the next operation

To reduce the calculation time due to the distortion calculation, we propose a quasi-optimal method which adapts both geometry and color quantization precision to intermediate meshes based on a simple analysis of geometry and color properties.

The procedure is quite similar to the method used for meshes without color attributes (Sect. 4.3). During the encoding process, the optimal number of geometry quantization bits q_G is estimated and if q_G is inferior to the current quantization precision Q_G , the decrease of geometry quantization precision is applied. In the same way, the decrease of color quantization is performed if the current value Q_C is superior to the estimated value q_C .

Surprisingly, for the geometry quantization, the same complexity measure used for meshes without color attributes (1) can be used, although a different distortion measure is employed. In Fig. 14, 89 values of K_G obtained by our optimal determination scheme using intermediate meshes of five colored models are plotted with the corresponding geometry quantization precisions.

By using the logarithm fitting curve and the least square method, we can now estimate q_G , the optimal geometry quantization precision:

$$q_G = \text{round}(-1.031 * \log(K_G) + 0.1477) \quad (4)$$

For the color quantization, we observe that one complexity measure cannot correctly determine the color quantization precision q_C for all intermediate meshes. Indeed, the determination of the optimal color quantization precision is more difficult to analyze than the geometry: contrary to the geometry, color coordinates are generally concentrated in a small portion of the color space, and the difference of color

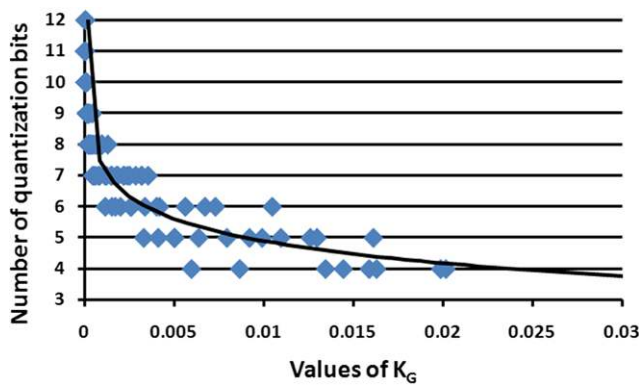


Fig. 14 Values of K_G with their corresponding optimal number of geometry quantization bits

between two adjacent vertices can also be very important. Moreover, in the vertex-based color representation, colors on a mesh surface depend on its geometric shape, since the color value of a point on a facet is obtained by interpolation using relative positions and colors of the facet vertices.

We can observe that the use of two complexity measures allows us to correctly determine the optimal color quantization precision q_C for all levels of detail: the first one is used to decide the initial and the final quantization precisions based only on information of the original mesh, and the second one is employed to determine adequate color quantization precision for all intermediate meshes.

To determine the initial and the final quantization precisions, we use the first complexity measure K_{C1} :

$$K_{C1} = \frac{\text{Mean color}}{\text{Max color}} \quad (5)$$

where *Mean color* is calculated as an average of the color differences of the vertices of all mesh edges and *Max color* is the maximum of these color differences.

In Figs. 15 and 16, we plot, respectively, the initial number of quantization bits and the final number of quantization bits obtained using our optimal determination method based on seven colored models. Using a polynomial curve fitting, we can now estimate the initial quantization precision $q_{C_{\text{init}}}$ by

$$q_{C_{\text{init}}} = \text{round}(-58.0 * K_{C1}^2 + 5.31 * K_{C1} + 7.81) \quad (6)$$

and the final quantization precision $q_{C_{\text{final}}}$ by

$$q_{C_{\text{final}}} = \text{round}(84.55 * K_{C1}^2 - 33.72 * K_{C1} + 7.82) \quad (7)$$

Now we need to determine the optimal color quantization precisions for the intermediate meshes. First, we consider the number of mesh elements in accordance with the concept of K_G . Then, we estimate the color complexity which reflects the color variation on the mesh surface by the ratio

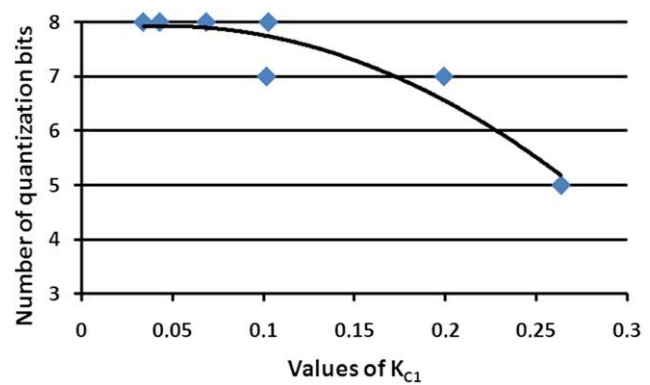


Fig. 15 Values of K_{C1} with their corresponding initial color quantization precisions

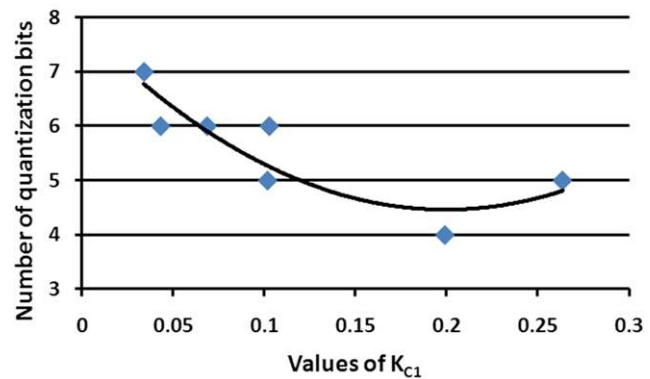


Fig. 16 Values of K_{C1} with their corresponding final color quantization precisions

between the mean value and the max value of the color differences. For the normalization, we also take the area of the mesh surface into account. To determine the optimal color quantization precision, we introduce a second complexity measure K_{C2} :

$$K_{C2} = \frac{\text{Mean color}}{\text{Max color}} \otimes \frac{\text{Surface area}}{\#V} \quad (8)$$

where *Surface area* is the sum of the mesh triangles' areas and $\#V$ denotes the number of mesh vertices.

Using the same seven colored models, we obtain values of K_{C2} with their corresponding quantization precisions. For the fitting, we use averaged values of K_{C2} for each quantization precision, as shown in Fig. 17 in order to improve the accuracy of the complexity measure.

By means of the logarithmic fitting curve and the least square method, we can estimate a near optimal quantization precision q_C for intermediate meshes:

$$q_C = \text{round}(-1.18 * \log(K_{C2}) + 0.33) \quad (9)$$

In our quasi-optimal algorithm, during the encoding we have to choose the next operation among decimation, decrease of geometry quantization precision, or decrease of

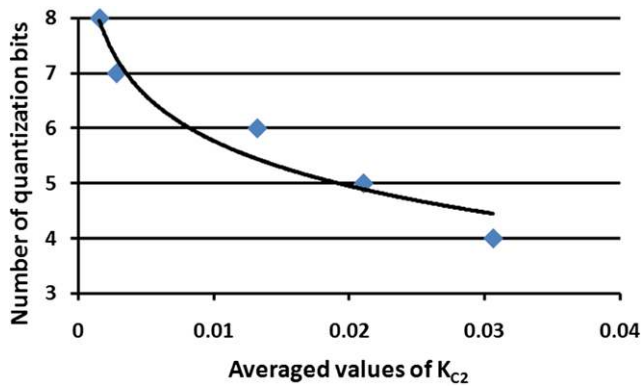


Fig. 17 Averaged values of K_{C2} with their corresponding optimal color quantization precisions

Algorithm 1 Our quasi-optimal method for R-D optimization at the encoding process

```

Calculate  $q_{C_{init}}$  (Eq. 6) and  $q_{C_{final}}$  (Eq. 7)
Decrease the current color quantization precision  $Q_C$ , until reaching  $q_{C_{init}}$ 
repeat
    Calculate  $q_C$  (Eq. 9) and  $q_G$  (Eq. 4)
    if  $Q_C > q_C$  AND  $Q_C > q_{C_{final}}$  then
        Decrease one bit of color quantization precision
    else if  $Q_G > q_G$  then
        Decrease one bit of geometry quantization precision
    else
        Decimate the current mesh
    end if
until Number of mesh elements > user-defined number

```

color quantization precision, at each iteration. Decimation is chosen when decreases of geometry and color quantization precision are not appropriate for the next operation. When decreases of geometry and color quantization levels are both appropriate, we have to make a decision between these two possibilities. In this case, our proposed solution is to choose the decrease of the color quantization level, since it is computed with less accuracy by using more equations (6), (7), and (9) than that of the geometry (4).

Our whole quasi-optimal algorithm is summarized in Algorithm 1.

5.6 Experimental results on meshes with color attributes

Table 2 shows the timings of our R-D optimization algorithm compared to the algorithm of Lee et al. (LLD 2010) [20] (one of our previous works). Q_G designates the number of geometry quantization bits. We can observe that our method using optimal determination requires much more calculation time due to the measurement of distortion with the tool of Roy et al. [27]. Our quasi-optimal approach

Table 2 Comparison of running timings in seconds

Model	#V	Q_G	LLD 2010	Our	
				Optimal	Quasi-optimal
Swirl	9216	10	0.2	22.1	0.4
Nefertiti	10013	10	0.3	35.6	1.2
Radiator	16002	10	0.4	49.7	0.5
Globe	36866	12	1.1	175.8	2.6
Gist-Monkey	67039	12	1.5	192.4	2.6
Envelope	125587	12	3.9	758.9	5.0
Renfort	190540	12	6.1	1495.9	6.8

allows us to considerably reduce the calculation time and makes our R-D optimization algorithm more effective.

In Figs. 18, 19, 20, and 21, respectively, we illustrate the R-D curves of the Radiator, the Enveloppe, the GIST-Monkey, and the Nefertiti models. The Radiator and the Enveloppe models are provided by the R&D division of EDF (Energie de France) and they represent outputs from thermal simulations. The distortion is obtained by measuring the maximum of two RMS distances using the metric of Roy et al. [27]. The Radiator and the Nefertiti are quantized using 10 bits, and the Enveloppe and the GIST-Monkey are quantized using 12 bits. The rate includes the connectivity, the geometry, and the color data. For all models, our algorithm based on optimal determination considerably improves the R-D performances in terms of the color deviation regarding to LLD 2010, especially at low resolutions, and our quasi-optimal determination technique gives very similar results.

In Fig. 20, we compare the R-D performance with the method of Yoon et al. [34]. In their work, they propose a prediction method for color encoding in the single-rate compression scheme. Hence, we adopt this prediction method into [20] (YKH(LLD 2010) in the figure) and also into our optimal algorithm (YKH(Our optimal) in the figure) for the comparison of the R-D trade-off. Since only the color encoding scheme is different, YKH(LLD 2010) and YKH(Our optimal) produce very similar results to those of, respectively, LLD 2010 and our optimal algorithm. We obtain the same results for other models.

In addition, the Enveloppe model was not used as the data set to establish quantization determination rules (see (4), (6), (7), (9)); hence it demonstrates once again the genericity of these rules. For the Radiator, a significant overhead of coding rate occurs due to the use of the color metric. As the Radiator model is initially regular, the forbidding of vertex removals during the mesh simplification raises the bit rates.

Even if our algorithm optimizes the R-D performance using a color-based metric, it also optimizes the geometry deviation during the whole transmission. The reason is that the color metric also takes into account the geometry information by using projection and barycentric interpolation to

Fig. 18 Rate/geometric and color distortion curve for the Radiator

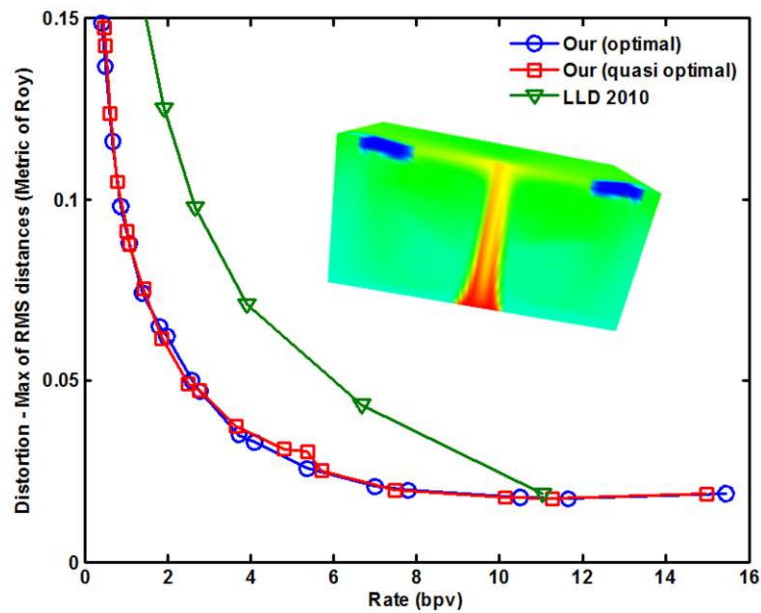
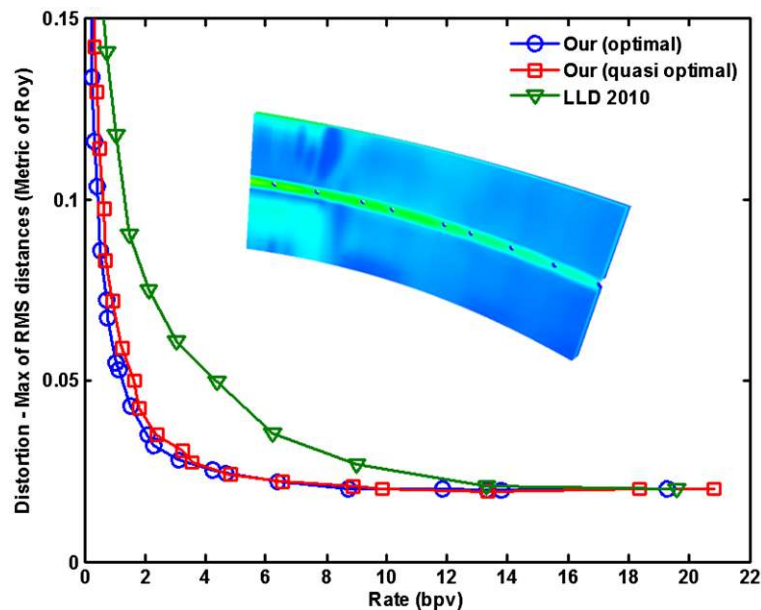


Fig. 19 Rate/geometric and color distortion curve for the Enveloppe



measure the distance between two meshes. We show the R-D curve obtained by the pure geometric metric, METRO [6] for the Nefertiti model in Fig. 22. We can see that our algorithm also outperforms LLD 2010 by far in terms of geometric deviation.

In Fig. 23, we show intermediate results of the Radiator model for similar bit rates of our algorithm compared to LLD 2010. For both algorithms, we use a volume-based geometric metric to preserve the initial shape. We can see that for all bit rates, our algorithm produces better results in terms of visual quality. The use of our metric allows us to more faithfully reconstruct color details adaptively to the

resolution levels. Note that our quasi-optimal algorithm delivers very similar results to those of our optimal algorithm.

6 Conclusion

In this paper, we have proposed a new lossless progressive compression technique for 3D meshes with color attributes. Based on the adaptation of the geometry and the color quantization precision to the complexity of intermediate meshes, an optimization of the R-D trade-off is performed. The next operation between decimation and decrease of quantization precision at the encoding process can

Fig. 20 Rate/geometric and color distortion curve for the GIST-Monkey

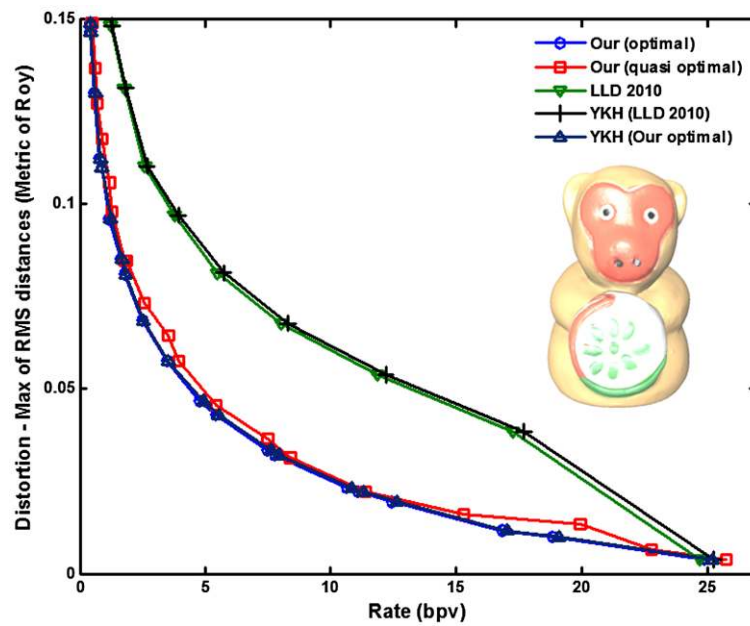
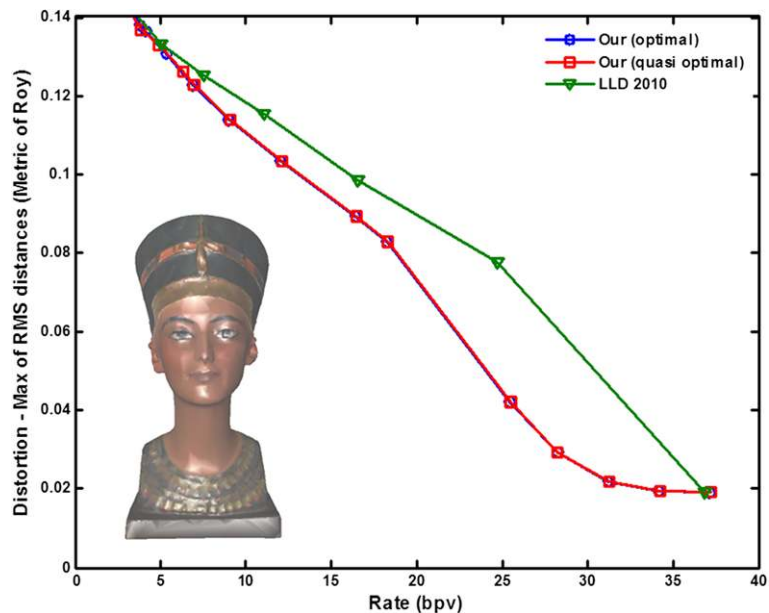


Fig. 21 Rate/geometric and color distortion curve for the Nefertiti



be determined either optimally using any mesh distortion measure, or quasi-optimally using an analysis of geometry and color complexity, with the goal of reducing the calculation time.

Experimental results show that the adaptive quantization and the use of our color metric allow improvement of the R-D performance in terms of color and geometry deviation for meshes with color attributes. In the case of meshes without attributes, our algorithm improves the R-D performance compared to the base algorithm [3] and can compete with the recent IPR algorithm [31]. One of the strong points of our approach is its genericity. Indeed the main idea of adapting

the quantization to the LODs can be easily adapted to other existing algorithms; moreover any distance can be used to optimize the R-D trade-off, even perceptual ones.

In our current framework, the adaptation of quantization precision is performed globally: vertices of a resolution level have the same number of quantization bits. In future works, we will improve our algorithm to allow the adaption of the quantization precision for each vertex during the reconstruction in order to further increase the R-D performance. We also plan to improve our quasi-optimal algorithm to more properly handle highly detailed objects by adapting its parameters to the amount of details.

Fig. 22 Rate/geometric distortion curve for the Nefertiti

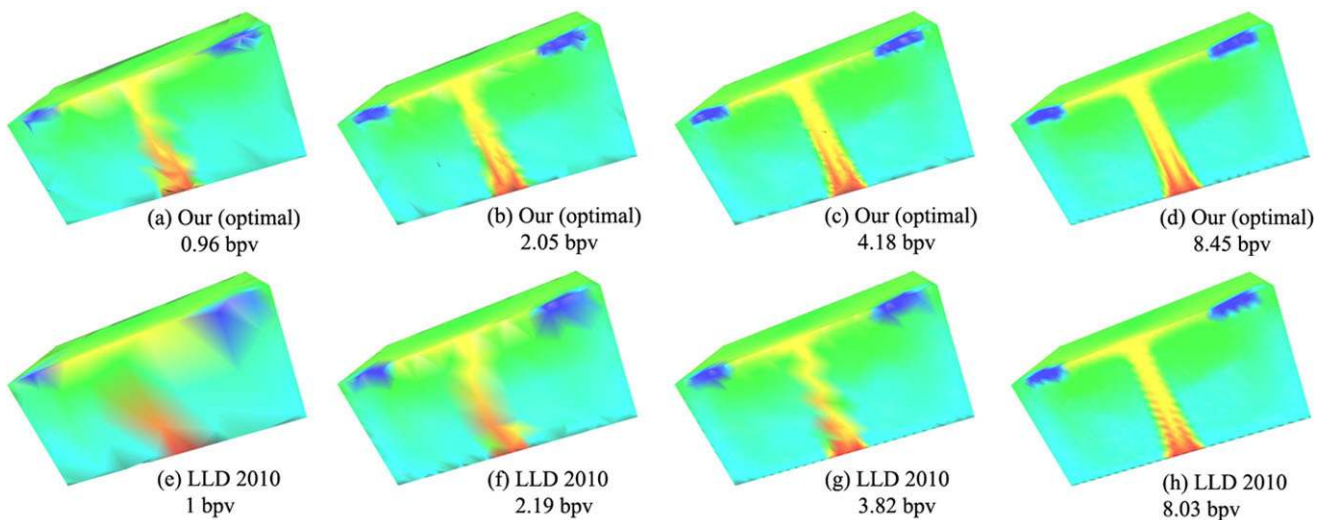
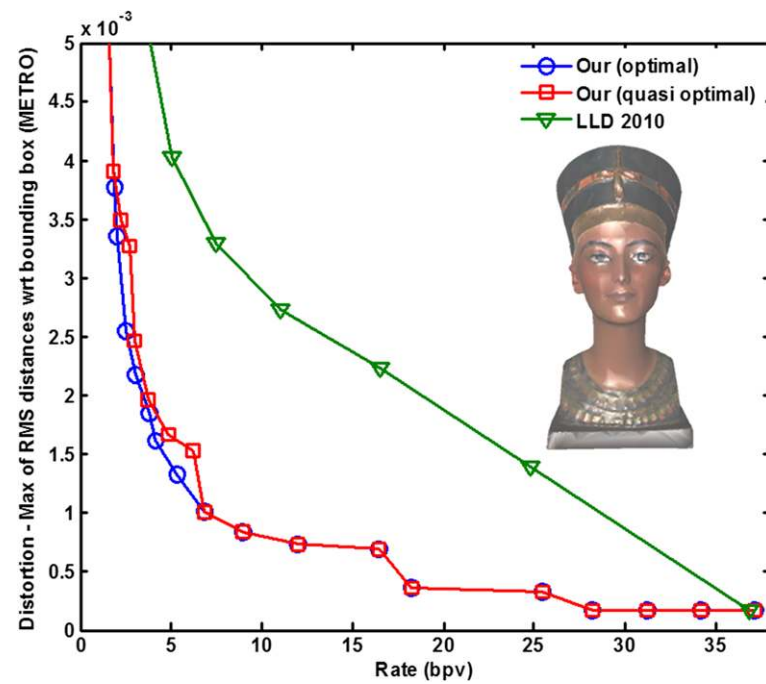


Fig. 23 Comparison of intermediate meshes for similar bit rates between our algorithm (optimal) and LLD 2010 [20]

Acknowledgements The authors thank the anonymous reviewers for their valuable comments which helped to improve the quality of the paper. This work has been supported by the French National Research Agency (ANR) through the COSINUS program (project COLLAVID n° ANR-08-COSI-003). The GIST-Monkey and Nefertiti models are courtesy of GIST in South Korea. The authors thank Sébastien Valette, Jingliang Peng, and Khaled Mamou for sending us the intermediate models from their works.

References

1. Ahn, J.H., Kim, C.S., Ho, Y.S.: Predictive compression of geometry, color and normal data of 3-D mesh models. *IEEE Trans. Circuits Syst. Video Technol.* **16**(2), 291–299 (2006)
2. Ahn, J.K., Lee, D.Y., Ahn, M., Kim, J.D.K., Kim, C., Kim, C.S.: Progressive compression of 3D triangular meshes using topology-based Karhunen-Loève transform. In: *IEEE International Conference on Image Processing*, pp. 3417–3420. IEEE Press, New York (2010)
3. Alliez, P., Desbrun, M.: Progressive compression for lossless transmission of triangle meshes. In: *Proceedings of SIGGRAPH*, pp. 195–202 (2001)
4. Alliez, P., Desbrun, M.: Valence-driven connectivity encoding for 3D meshes. *Comput. Graph. Forum* **20**(3), 480–489 (2001)
5. Bajaj, C.L., Pascucci, V., Zhuang, G.: Single resolution compression of arbitrary triangular meshes. In: *Proceedings of the Conference on Data Compression*, pp. 247–256 (1999)

6. Cignoni, P., Rocchini, C., Scopigno, R.: Metro: measuring error on simplified surfaces. *Comput. Graph. Forum* **17**(2), 167–174 (1998)
7. Cirio, G., Lavoué, G., Dupont, F.: A framework for data-driven progressive mesh compression. In: *International Conference on Computer Graphics Theory and Applications (GRAPP)* (2010)
8. Cohen-Or, D., Levin, D., Remez, O.: Progressive compression of arbitrary triangular meshes. In: *IEEE Visualization 99*, pp. 67–72 (1999)
9. Deering, M.: Geometry compression. In: *Proceedings of SIGGRAPH*, vol. 95, pp. 13–20. ACM Press, New York (1995)
10. Gandoin, P., Devillers, O.: Progressive lossless compression of arbitrary simplicial complexes. *ACM Trans. Graph. (TOG)* **21**(3), 372–379 (2002)
11. Gumhold, S., Straßer, W.: Real time compression of triangle mesh connectivity. In: *Proceedings of SIGGRAPH*, p. 140. ACM Press, New York (1998)
12. Guskov, I., Vidimče, K., Sweldens, W., Schröder, P.: Normal meshes. In: *Proceedings of SIGGRAPH*, pp. 95–102 (2000)
13. Hoppe, H.: Progressive mesh. In: *Proceedings of SIGGRAPH*, vol. 96, pp. 99–108. (1996)
14. Isenburg, M., Lindstrom, P., Snoeyink, J.: Lossless compression of predicted floating-point geometry. *Comput. Aided Des.* **37**(8), 869–877 (2005)
15. Karni, Z., Bogomjakov, A., Gotsman, C.: Efficient compression and rendering of multi-resolution meshes. In: *IEEE Visualization*, pp. 347–354 (2002)
16. Khodakovsky, A., Guskov, I.: Compression of normal meshes. In: *Geometric Modeling for Scientific Visualization*, pp. 189–206. Springer, Berlin (2004)
17. Khodakovsky, A., Schröder, P., Sweldens, W.: Progressive geometry compression. In: *Proceedings of SIGGRAPH*, pp. 271–278 (2000)
18. King, D., Rossignac, J.: Optimal bit allocation in 3D compression. *J. Comput. Geom. Theory Appl.* **14**, 91–118 (1999)
19. Lee, H., Lavoué, G., Dupont, F.: Adaptive coarse-to-fine quantization for optimizing rate-distortion of progressive mesh compression. In: *Vision, Modeling, and Visualization (VMV)*, pp. 73–81. (2009)
20. Lee, H., Lavoué, G., Dupont, F.: New methods for progressive compression of colored 3D Mesh. In: *International Conference on Computer Graphics, Visualization and Computer Vision (WSCG)*, pp. 199–206. (2010)
21. Mamou, K., Dehais, C., Chaieb, F., Ghorbel, F.: Shape approximation for efficient progressive mesh compression. In: *IEEE International Conference on Image Processing*, vol. 1, pp. 3425–3428. IEEE Press, New York (2010)
22. Pajarola, R., Rossignac, J.: Compressed progressive meshes. *IEEE Trans. Vis. Comput. Graph.* **6**(1), 79–93 (2000)
23. Payan, F., Antonini, M.: An efficient bit allocation for compressing normal meshes with an error-driven quantization. *Comput. Aided Geom. Des.* **22**(5), 466–486 (2005)
24. Peng, J., Huang, Y., Kuo, C.C.J., Eckstein, I., Gopi, M.: Feature oriented progressive lossless mesh coding. *Comput. Graph. Forum* **29**(7), 2029–2038 (2010)
25. Peng, J., Kuo, C.: Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. *ACM Trans. Graph. (TOG)* **24**(3), 616 (2005)
26. Rossignac, J.: Edgebreaker: connectivity compression for triangle meshes. *IEEE Trans. Vis. Comput. Graph.* **5**(1), 47–61 (1999)
27. Roy, M., Fofou, S., Truchetet, F.: Mesh comparison using attribute deviation metric. *Int. J. Image Graph.* **4**(1), 127–140 (2004)
28. Taubin, G., Guézic, A., Horn, W., Lazarus, F.: Progressive forest split compression. In: *Proceedings of SIGGRAPH*, pp. 123–132 (1998)
29. Taubin, G., Rossignac, J.: Geometric compression through topological surgery. *ACM Trans. Graph. (TOG)* **17**(2), 84–115 (1998)
30. Touma, C., Gotsman, C.: Triangle mesh compression. In: *Graphics Interface 98 Conference Proceedings*, pp. 26–34 (1998)
31. Valette, S., Chaine, R., Prost, R.: Progressive lossless mesh compression via incremental parametric refinement. *Comput. Graph. Forum* **28**(5), 1301–1310 (2009)
32. Valette, S., Gouaillard, A., Prost, R.: Compression of 3D triangular meshes with progressive precision. *Comput. Graph.* **28**(1), 35–42 (2004)
33. Valette, S., Prost, R.: Wavelet-based progressive compression scheme for triangle meshes: wavemesh. *IEEE Trans. Vis. Comput. Graph.* **10**(2), 123–129 (2004)
34. Yoon, Y.S., Kim, S.Y., Ho, Y.S.: Color data coding for three-dimensional mesh models considering connectivity and geometry information. In: *IEEE International Conference on Multimedia and Expo*, pp. 253–256 (2006)



Ho Lee received his engineering degree (2007) and his M.S. degree (2008), both in electrical engineering, from Institut National des Sciences Appliquées de Lyon, France. He is currently a Ph.D. candidate at the LIRIS Laboratory of Université Claude Bernard Lyon 1, France. His research interests include progressive compression and watermarking for 3D models.



Guillaume Lavoué received his engineering degree in signal processing and computer science from CPE Lyon (2002), his M.Sc. degree in image processing from the University Jean Monnet, St-Etienne (2002), and his Ph.D. degree in computer science from the University Claude Bernard, Lyon, France (2005). Since September 2006 he has been an associate professor at the French engineering university INSA of Lyon, in the LIRIS Laboratory (UMR 5205 CNRS). His research interests include 3D model analysis and processing, including compression, watermarking, perception, and 2D/3D recognition.



Florent Dupont received his B.S. and M.S. degrees in 1990 and his Ph.D. in 1994 from Institut National des Sciences Appliquées de Lyon, France. He became an associate professor in 1998. He is now a professor and leads the Multiresolution, Discrete and Combinatorial Models (M2DisCo) team of the LIRIS Laboratory at the Université Claude Bernard Lyon 1, France. His technical research concerns 3D digital image processing and computational geometry.