Wavelet-Based Multiple Description Coding of 3-D Geometry

Andrey Norkin[a], M. Oguz Bici[b], Gozde Bozdagi Akar[b], Atanas Gotchev[a], and Jaakko Astola[a]

[a]Institute of Signal Processing, Tampere University of Technology, Tampere, Finland;
[b]Middle East Technical University, Ankara, Turkey.

ABSTRACT

In this work, we present a multiple description coding (MDC) scheme for reliable transmission of compressed three dimensional (3-D) meshes. It trades off reconstruction quality for error resilience to provide the best expected reconstruction of 3-D mesh at the decoder side. The proposed scheme is based on multiresolution geometry compression achieved by using wavelet transform and modified SPIHT algorithm. The trees of wavelet coefficients are divided into sets. Each description contains the coarsest level mesh and a number of tree sets coded with different rates. The original 3-D geometry can be reconstructed with acceptable quality from any received description. More descriptions provide better reconstruction quality. The proposed algorithm provides flexible number of descriptions and is optimized for varying packet loss rates (PLR) and channel bandwidth.

Keywords: Multiple description coding, 3-D geometry, meshes, SPIHT, wavelets, error resilience.

1. INTRODUCTION

3-D models are widely used nowadays. The most common representation for 3-D objects is a surface based triangle mesh.[1] Achieving high quality models requires large number of triangles. Therefore, compressing mesh data is necessary due to storage space and bandwidth limitations. A great number of 3-D mesh compression schemes have been proposed in the literature.[2] In multimedia applications such as virtual presence, internet computer games, e-commerce, tele-medicine, object-based video compression, 3DTV, 3-D meshes need to be communicated in a networked environment. However, in a typical network, packets may be lost or delayed because of congestions and buffer overflow. There are several error-resilient coding schemes for meshes in the literature. In the approach of Yan et al.,[3] error resilience is achieved by segmenting the mesh and transmitting each segment independently. At the decoder, these segments are stitched using the joint-boundary information which is considered the most important. However, this method does not provide a coarse-to-fine representation of the model. AlRegib et al.[4] proposed to assign optimal error correcting codes to layers of progressively coded 3-D mesh. This method is scalable with respect to both channel bandwidth and channel packet loss rate. It can be applied to any progressive compression scheme, which produces a hierarchical bitstream. However, if a coarse layer cannot be recovered due to packet losses, all the refinement layers become useless and decoding process terminates.

An alternative to layered progressive mesh coding is Multiple Description Coding (MDC). MDC is a source-channel coding of information, which can be represented with different levels of quality. The source is encoded into several bitstreams (i.e. multiple descriptions to be transmitted via independent channels). In the receiver, source can be reconstructed from any single bitstream at lower yet acceptable quality. Higher quality is achieved if more bitstreams are received. Representing the source with different levels of quality makes MDC similar to layered coding. However, while the latter requires correct reception of the base layer in order for enhancement layers to be useful, the former can reconstruct the source from any subset of bitstreams.[5]

Further author information: (Send correspondence to A.N.)
A.N.: E-mail: andrey.norkin@tut.fi
M.O.B.: E-mail: mobici@eee.metu.edu.tr
G.B.A.: E-mail: bozdagi@eee.metu.edu.tr
A.G.: E-mail: atanas.gotchev@tut.fi
J.A.: E-mail: jaakko.astola@tut.fi

Although MDC of images, video and audio has been extensively studied, little research has been done for MDC of 3-D geometry. In Ref.,[6] multiple descriptions are generated by splitting the mesh geometry into submeshes and including the whole connectivity information in each description. Another approach[7] applies multiple description scalar quantization (MDSQ) to wavelet coefficients of a multiresolution compression scheme. The obtained two sets of coefficients are then independently compressed by the SPIHT coder. However, in those MDC schemes, descriptions are created with heuristic methods and no optimal solutions are proposed for varying network conditions.

In this paper, we propose a novel approach for MDC of 3D-meshes. Our MD coder is based on Progressive Geometry Compression (PGC) scheme,[8] where the input model is remeshed to obtain a semi-regular mesh that approximates the geometry of the original mesh. Then, wavelet transform is applied. The output of wavelet transform is the coarsest level mesh and wavelet coefficients that form edge-based trees. These edge-based trees are coded with SPIHT algorithm. To provide robustness to channel errors, wavelet coefficient trees are grouped into several sets, which are independently coded. Two strategies of forming tree sets are considered in this paper. Those sets are packetized into multiple descriptions in such a way that each description contains one tree set that is coded with higher rate and several redundant tree sets coded with lower rates.

2. THE PROPOSED SCHEME

Our MDC coder is based on PGC scheme,[8] which is a progressive compression scheme for arbitrary topology, highly detailed and densely sampled meshes arising from geometry scanning. The original model in PGC is remeshed to have a semi-regular structure. The remeshing is done in such a way that the error caused by remeshing is smaller than the estimated discretization error. The obtained semi-regular mesh undergoes a loop-based or butterfly-based wavelet decomposition to produce a coarsest level mesh and wavelet coefficients. For the reason of its irregular structure, coarsest level connectivity is coded by Touma and Gotsman (TG) coder.[9] The geometry data of the coarsest mesh is uniformly quantized and added to bitstream. Wavelet coefficient (edge-based) trees are coded with SPIHT algorithm.[10]

In order to obtain multiple descriptions from wavelet coefficients, we adapted the ideas from MD image coding.[11,12] Suppose that the coder generates N descriptions. In our coder (Fig. 1), wavelet coefficient trees are split into several sets W_i, $i = 1 \dots N$ and coded by SPIHT algorithm at high bitrate. These sets are included in the descriptions in a following way. Each description contains M copies of different tree sets ($M \leq N$). Namely, *Description* i contains one set W_i coded at rate $R_{i,0}$ and $M-1$ sets of redundant trees W_j, $j \neq i$. These $M-1$ tree sets represent coding redundancy and are coded at lower rates than $R_{i,0}$. Rates $R_{i,j}$ are obtained as a result of optimization explained in Section 3.

If all the descriptions are received, only the trees coded with higher rates $R_{i,0}$ are used for reconstruction. Redundant copies are used when the descriptions that contain the higher-rate copies of coded tree sets are lost. Thus, if some descriptions are lost, most important parts of the original trees in those descriptions are recovered from their copies coded at lower rates. The compressed coarsest mesh representation C with rate R_C is included in every description to facilitate the inverse wavelet transform even if only one description is received. Duplicating coarsest mesh C also increases coding redundancy.

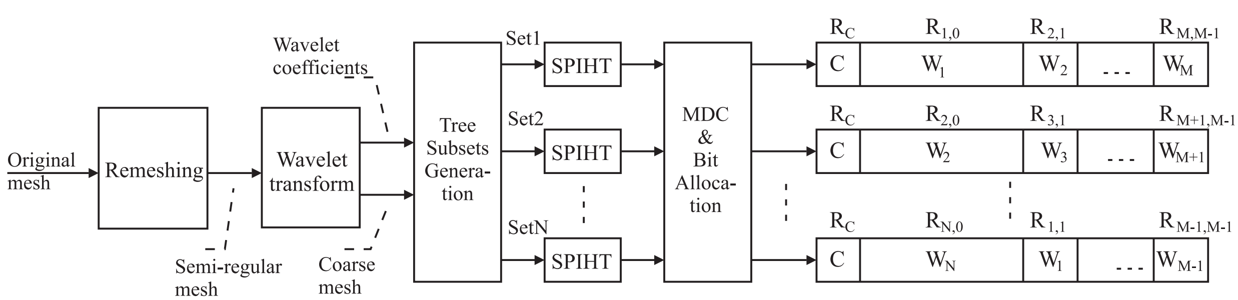
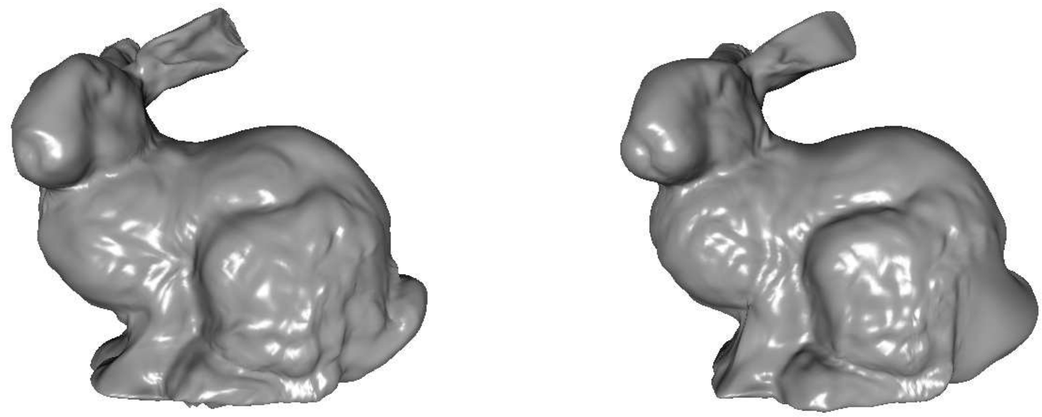


Figure 1. The encoder scheme.



(a) PSNR= 50.73 dB (b) PSNR= 48.51 dB

Figure 2. Reconstruction from one description for different types of tree grouping. (a) Spatially disperse grouping; PSNR= 50.73 dB. (b) Spatially close grouping, group size is 10; PSNR= 48.51 dB.

The way of grouping coefficient trees into sets is particularly important, since different sets are reconstructed with different quality in case of one description loss. Therefore, 3-D mesh locations corresponding to different tree sets will have different quality. In our experiments, we have tried two types of tree set grouping: grouping closely located trees together and grouping spatially disperse trees.

To perform grouping of spatially dispersed or spatially close trees into sets, ordering of first-level wavelet coefficients (roots of wavelet trees) is performed. This ordering is performed together with the coarsest mesh vertices with the algorithm proposed in Ref.[13,14] This algorithm provides ordering of vertices, which has good locality and continuity properties. Then, the desired type of grouping is obtained by sampling a one-dimensional array, which is the output of the ordering algorithm.[13,14] Disperse grouping is obtained by sampling the array in a round-robin fashion. Spatially close grouping is obtained by assigning groups of successive vertices with the chosen group size to the same set.

We have found that visual quality is better by grouping spatially close trees. Grouping spatially disperse trees provides annoying artifacts when reconstructing from only one description. This is illustrated in Fig. 2. Model Bunny is encoded into four descriptions and optimized for PLR = 15%. Two grouping strategies are compared: grouping spatially disperse trees and grouping spatially close trees with group size 10. The model is reconstructed from the *Description 1* alone. One can see that although grouping disperse disperse trees achieves lower objective distortion than grouping close trees, it produces annoying visual artifacts. Therefore, in the following, we use spatially close grouping with group size 10.

3. OPTIMIZING BIT ALLOCATION

Redundancy of the proposed algorithm is determined by the number of redundant tree copies and their rates, and the coarsest mesh size. Bit allocation problem has to minimize expected distortion at the decoder subject to probability of packet loss P and the target bit budget. We use a simple channel model where probability of packet loss P for each packet is the same and independent of previous channel events. In the following, we assume for simplicity that one packet corresponds to one description. If the description has to be fragmented into different packets, probability of the description loss P can easily be found from PLR.

Suppose that N descriptions are generated. Then, coefficient trees are split into N tree sets, and M copies of tree sets are included in one description ($M \leq N$). Given P, it is easy to determine for each copy of the tree set the probabilities P_i, $i = 0, \dots M$ that this copy is used for reconstruction where P_0 is the probability of using the full-rate copy of the tree set, and P_M is the probability of not receiving any copy of the tree set. Probabilities P_i

can easily be found from P and the packetization strategy. Thus, we have to minimize the expected distortion

$$E[D] = \sum_{i=1}^{N} \sum_{j=0}^{M} P_j D_{ij}(R_{ij}), \tag{1}$$

where D_{ij} is the distortion incurred by using j-th copy of a tree set i and R_{ij} represents bits spent for j-th copy of i-th tree set. Optimization is performed under following bitrate constraints

$$\sum_{i=1}^{N} \sum_{j=0}^{M} R_{ij} + NR_C \leq R, \tag{2}$$

where R is the target bitrate and NR_C is the rate of the coarsest mesh. The rate of the coarsest mesh is chosen constant with geometry information quantized to 14 bitplanes.

Optimization of bit allocation requires computation of $D(R)$ function for every allocation step. Calculation of $D(R)$ is a computationally expensive operation. However, each tree set contributes to total distortion D. Since each tree set corresponds to some separate location on the mesh surface (defined by the root edge) in grouping spatially close trees, the distortions corresponding to separate tree sets can be considered additive. Therefore, distortion-rate (D-R) curve $D_i(R_i)$ for each coefficient tree set is obtained in advance. Calculations of $D_i(R_i)$ are performed only once, before the optimization algorithm is used for the first time. Then, D-R are saved and can be used every time in bit allocation algorithm for new values of R and P.

Optimization is performed with generalized Breiman, Friedman, Olshen, and Stone (BFOS) algorithm.[15] BFOS algorithm first allocates high rate for each copy of the tree set. Then, the algorithm consequently deallocates bits from the sets where $D(R)$ curves shows the lowest decay at allocated bitrate. This process stops when bit budget constraints are satisfied. In case optimization brings zero rates for some redundant trees copies, these copies are not included in the descriptions. Simulations show that bit-allocation algorithm exhibits the desired behavior. The higher is the packet loss rate, the more bits are allocated to redundant copies providing more robustness to packet losses.

4. COMPLEXITY ISSUES AND DISTORTION-RATE FUNCTION MODELING

There is no immediate objective distortion metric in 3D meshes like mean-square error in images. One of the most popular and robust objective distortion metric is L^2 distance between two surfaces. L^2 distance between two surfaces X and Y is defined as

$$d(X,Y) = \left(\frac{1}{area(X)} \int_{x \in X} d(x,Y)^2 dx \right)^{1/2}. \tag{3}$$

Since the distance is not symmetric, it is symmetrized by taking maximum of $d(X,Y)$ and $d(Y,X)$. Metro tool[16] approximates this distance by sampling vertices, edges, and triangles and taking root mean square value of shortest distances from points in X to surface Y. Calculation of these distances is an expensive operation. Therefore, long off-line computations are needed to obtain operational D-R curves that are used in bit allocation.

To increase the speed of calculating distortion, we use D-R function modeling[17] presented for wavelet-based coding of images. In our experiments, we use a Weibull model.[17] We have found that Weibull model can approximate well D-R function of wavelet coefficient tree sets. The Weibull model is

$$D(R) = a - be^{-cR^d}, \tag{4}$$

where real numbers a, b, c, and d are parameters, which depend on the D-R characteristics of the source and the bitstream. As there are four parameters in this model, $D(R)$ curve can be found by using at minimum four points that considerably decreases the amount of computations. This model can approximate both L^2 and PSNR curves. To fit Weibull model to D-R samples, we use nonlinear least-squares regression.

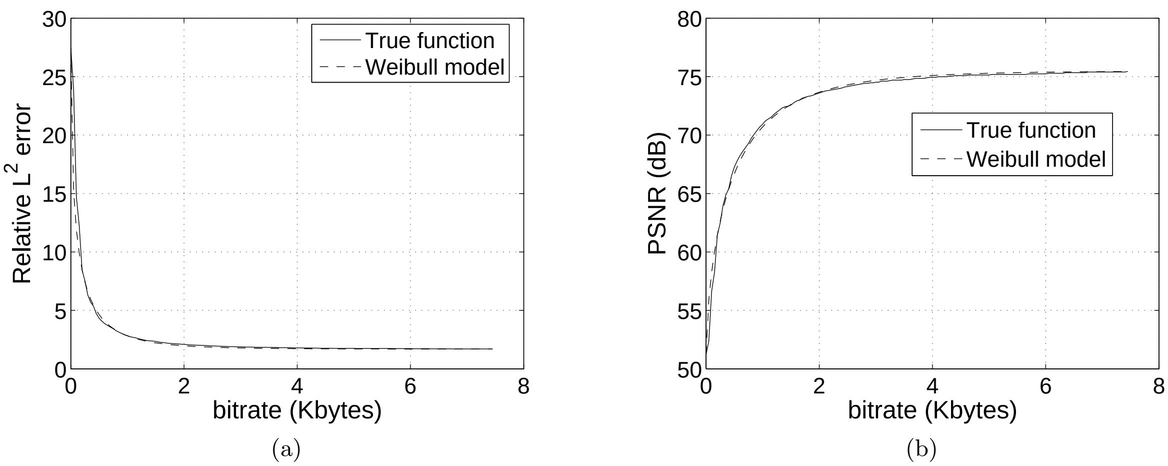


Figure 3. Comparison between the Weibull model (4 samples) and operational D-R curve (relative L^2 error) for the first set of wavelet coefficient trees for *Bunny* model. (a) Relative L^2 error in units of 10^4; (b) PSNR.

Fig. 3 compares true operational D-R curves and their Weibull models. The Weibull models are $D(R) = 0.000169 - 0.002578e^{-0.8134R^{0.52805}}$ and $D(R) = 75.48 + 24.264e^{0.0153R^{0.675}}$ for L^2-distance and PSNR respectively.

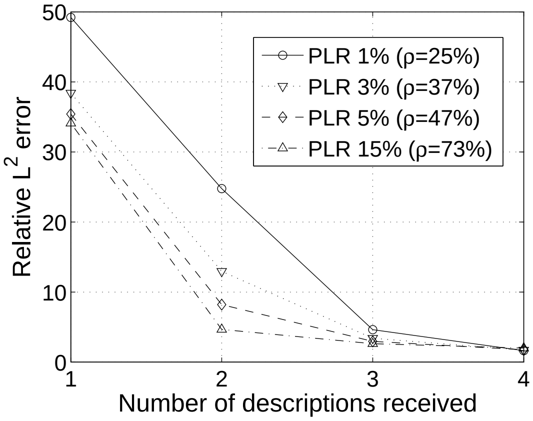
One can see that the model closely approximates the real data. Moreover, the model has a nice feature of convexity, which is desirable for bit allocation algorithm. As the model needs only four samples to model the D-R curve, only $4N$ D-R samples are computed to generate D-R curves for N sets.

5. SIMULATION RESULTS

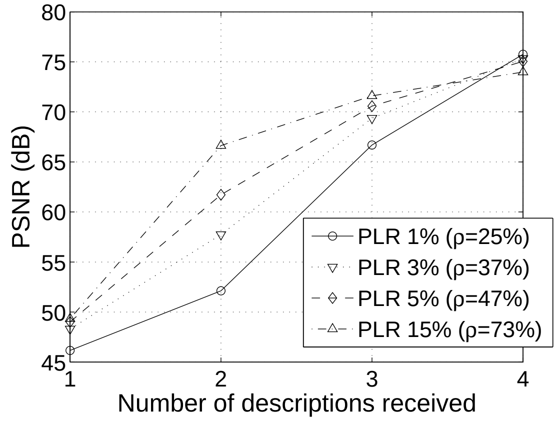
In this section, we present simulation results for test models *Bunny* and *Venus head*. The reconstruction distortion is relative L^2 error, which is calculated with Metro tool.[16] Relative error is calculated by dividing L^2 distance to the original mesh bounding box diagonal. The error is shown in the figures in units of 10^{-4}. We also provide same numbers in PSNR scale where PSNR $= 20 \log_{10} peak/d$, *peak* is the bounding box diagonal, and d is the L^2 error. When all descriptions are lost and no reconstruction is possible, we calculate the distortion as the L^2 distance between the surface of the original mesh and a single point with coordinates $(0, 0, 0)$.

In the experiments, we compare three coders. The first coder is the one proposed in this paper, which is named Tree-based Mesh MDC (TM-MDC). The second coder is a simple MDC coder where the 3-D model is coded into four descriptions. Each description contains the coarsest mesh and one set of wavelet coefficient trees. The sets of coefficient trees in both coders are formed from spatially close groups of trees of size 10. This coder is the same as TM-MDC optimized for $P = 0$ (for $P = 0$, no redundant trees are included in the descriptions). The third coder is unprotected SPIHT. The packetization for unprotected SPIHT is performed in the following way. The output bitstream of PGC coder is divided into N parts of equal size, where N is the number of descriptions in the MD coder that unprotected SPIHT is compared to. PGC produces the embedded bitstream. Thus, the received part can be used for reconstruction if all the packets containing earlier parts of bitstream have been received. For example, if parts one, two, and four are received, only parts one and two are used for reconstruction. If part one is lost, no reconstruction is available.

Fig. 4 and Fig. 5 show the average distortion for reconstruction for the proposed algorithm from different number of received descriptions for models *Bunny* and *Venus head*. Model *Bunny* is coded at 22972 Bytes (5743 Bytes per each description) and model *Venus head* at 24404 Bytes (6101 Bytes per each description). The curves are generated for TM-MDC. Bit allocation is performed for each value of P. The corresponding value of redundancy ρ is given in brackets. One can see that the coder optimized for $P = 1\%$ shows the best performance when all the descriptions are received while the coder optimized for $P = 15\%$ shows the best results for reconstruction from one, two, and three descriptions.

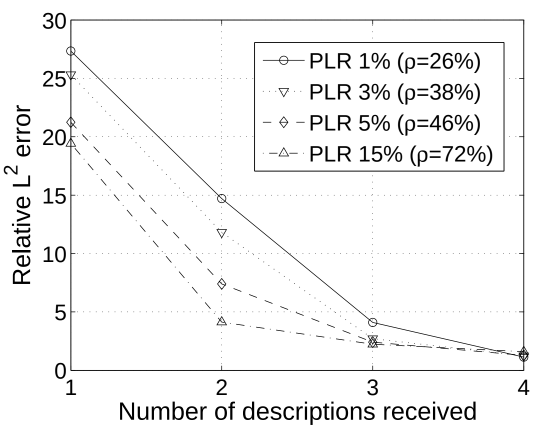


(a)

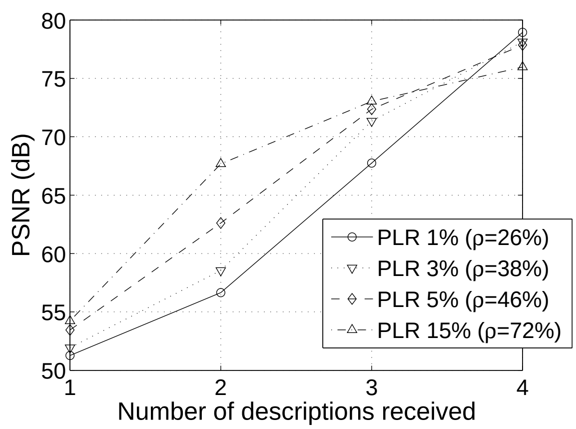


(b)

Figure 4. Reconstruction of *Bunny* model from different number of descriptions. The results are given for bit allocations for different packet loss rates (PLR). The redundancy ρ is given in brackets. (a) Relative L^2 error in units of 10^4; (b) PSNR.



(a)



(b)

Figure 5. Reconstruction of *Venus head* model from different number of descriptions. The results are given for bit allocations for different packet loss rates (PLR). The redundancy ρ is given in brackets. (a) Relative L^2 error in units of 10^4; (b) PSNR.

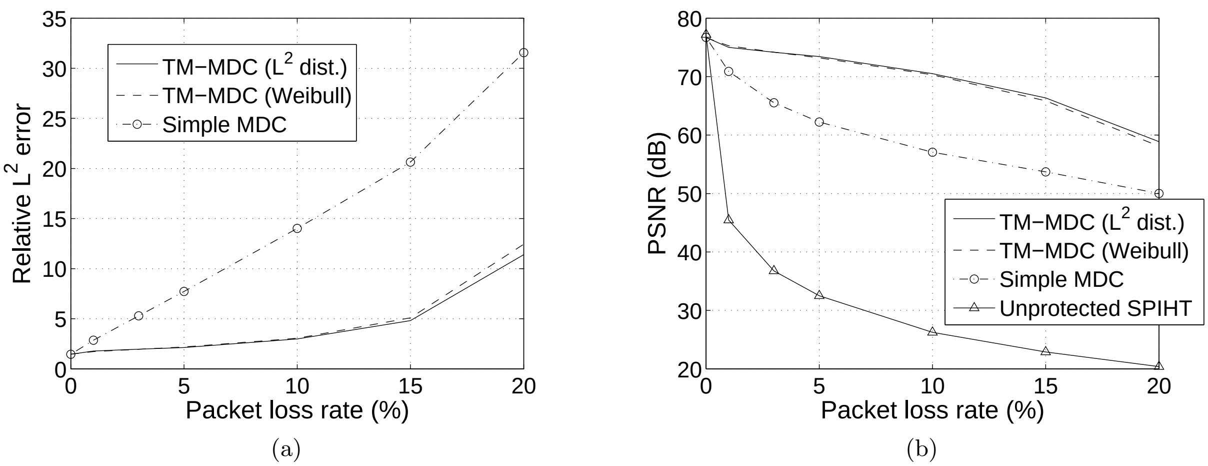


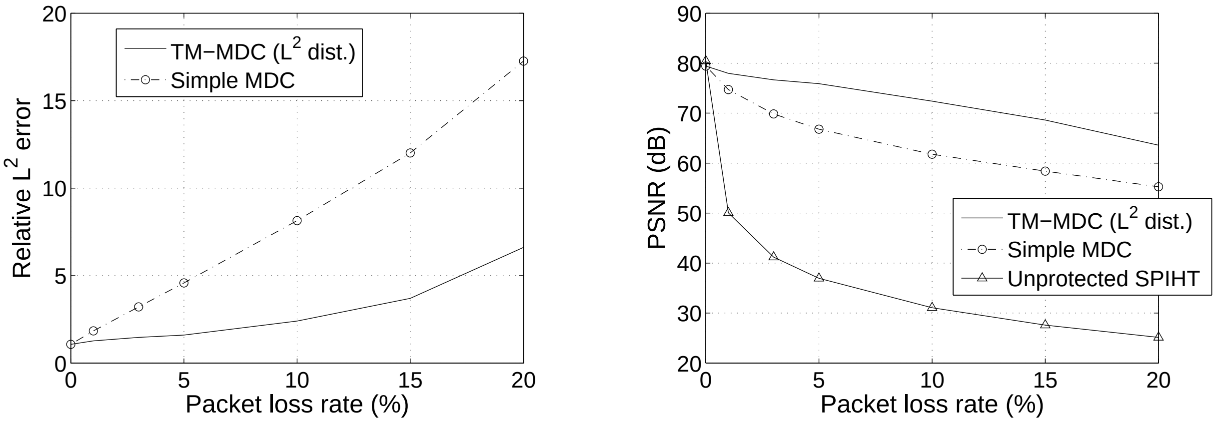
Figure 6. Model *Bunny*. Comparison of network performance of the proposed TM-MDC with a simple MDC scheme and unprotected SPIHT. The results for TM-MDC with D-R curve modeling are given as TM-MDC (Weibull). (a) Relative L^2 error in units of 10^4; (b) PSNR.

Fig. 6 and Fig. 7 compare the network performance of the proposed TM-MDC, the simple MD coder, and unprotected SPIHT. The results are calculated for $P = 0, 1, 3, 5, 10, 15, 20\%$. In TM-MDC coder, bit allocation is optimized for each P. For simple MDC coder, the value of redundancy is always fixed. These figures also compare performance of TM-MDC using "real" operational D-R curves for bit allocation and TM-MDC using D-R function modeling to increase the speed of data collection. For each P, we have performed 100000 experiments (simulations of packet losses) and calculated the average distortion. For $P = 0$, TM-MDC and simple MD coder show the same performance because the optimized for $P = 0$ TM-MDC coder and simple MD coder are in fact the same coder. Unprotected SPIHT shows slightly better performance than MD coders in the error-free environment ($P = 0$). However, for higher values of P, performance of simple MDC coder and unprotected SPIHT dramatically decreases while the reconstruction quality of TM-MDC shows mild degradation. For $P = 20\%$, the optimized TM-MDC coder shows PSNR 10 dB higher than the simple MD coder and 30 to 40 dB higher than unprotected SPIHT. Because of big difference in distortion values between MD coders and unprotected SPIHT, the results for the latter one are shown only in PSNR scale. TM-MDC using Weibull model for D-R function shows almost as good performance as TM-MDC using real operational D-R curves. Thus, D-R curve modeling can be used to decrease the amount of computations at preparatory stage without significant decrease in network performance.

Fig. 8 shows the average distortion for reconstruction from different number of received descriptions for model *Bunny* encoded into 8 descriptions at total 25944 Bytes. The curves are generated for TM-MDC coder with bit allocation for $P = 5, 10\%$. The value of redundancy ρ is given in brackets. TM-MDC is compared to unprotected SPIHT. One can see that unprotected SPIHT achieves higher PSNR than TM-MDC only in case when all the descriptions are received. When less than eight descriptions are received, TM-MDC exhibits much higher PSNR.

Fig. 9 shows the trade-off between the central distortion D_0 and mean-side distortion $(D_1 + D_2)/2$. The plot is presented for Bunny model coded into two descriptions at total 21486, 16486, and 11486 Bytes.

Figures 10 and 11 show visual reconstruction results. In Fig. 10, model *Bunny* is encoded into four descriptions with redundancy $\rho = 63\%$ at total 22972 Bytes and reconstructed from one, two, tree, and four description. One could see that even the reconstruction from one description provides acceptable visual quality. Fig. 11 shows the visual reconstruction results for *Venus head* model, encoded into four descriptions with redundancy $\rho = 53\%$ at total 24404 Bytes. Both models are encoded with grouping spatially close trees with group size 10.



(a) (b)

Figure 7. Model *Venus head*. Comparison of network performance of the proposed TM-MDC with a simple MDC scheme and unprotected SPIHT. (a) Relative L^2 error in units of 10^4; (b) PSNR.

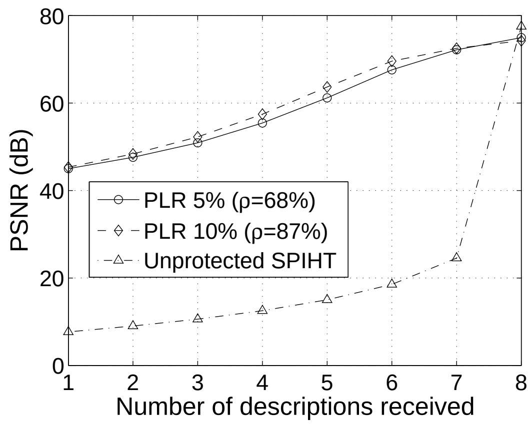


Figure 8. Model *Bunny* encoded into 8 descriptions at total 25944 Bytes. Reconstructed from different number of descriptions. Compared to unprotected SPIHT.

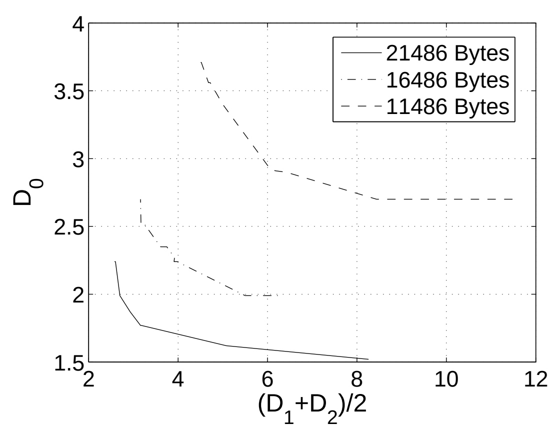


Figure 9. Model *Bunny* encoded into two descriptions at total 21486, 16486, and 11486 Bytes. Central vs mean-side distortion.

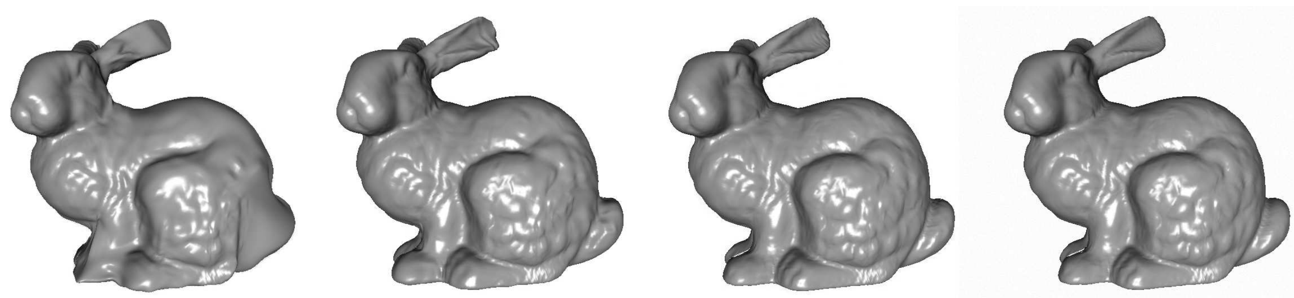


Figure 10. Reconstruction of *Bunny* model from (left to right): one description (48.36 dB), two descriptions (63.60 dB), three descriptions (71.44 dB), four descriptions (74.33 dB).

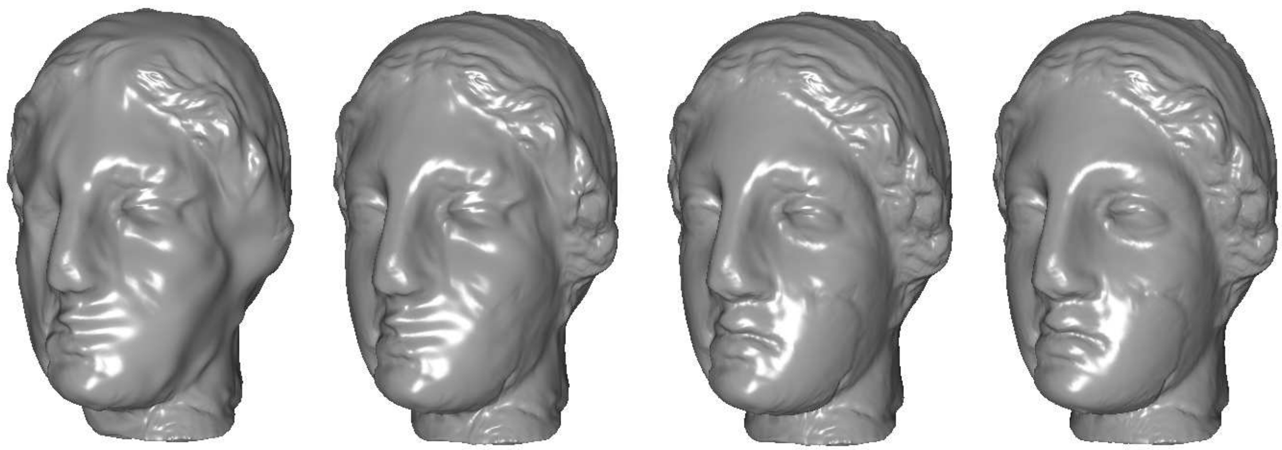


Figure 11. Reconstruction of *Venus head* model from (left to right): one description (53.97 dB), two descriptions (65.18 dB), three descriptions (72.51 dB), four descriptions (77.08 dB).

6. CONCLUSIONS AND FUTURE WORK

We have proposed a novel approach for MDC of 3-D meshes. Our MD coder is based on PGC scheme.[8] Wavelet transform is applied to the semi-regular remeshed model, and wavelet coefficient trees are coded with SPIHT algorithm. Our algorithm exploits redundancy in a form of redundant partially-coded trees of wavelet coefficients and a duplicated coarsest mesh. Two strategies of grouping the trees were studied, and spatially close grouping was chosen as the one providing better visual quality. To decrease the amount of computations at the preparatory stage, D-R function modeling can be exploited with the price of only slight decrease in network performance. The algorithm is capable of providing flexible number of descriptions and is optimized for varying packet loss rate. Graceful degradation of quality is achieved in presence of increasing packet loss rate. The reconstruction results show good visual perception quality.

The plans for future in this direction include optimization of the coarse mesh bit allocation and partitioning coarsest mesh vertices into different sets in order to further decrease the amount of redundancy. We also plan to consider more complicated channel model and packetization strategy which better emulate the behavior of a real network and further investigate tree grouping strategies.

7. ACKNOWLEDGEMENTS

This work is supported by EC within FP6 under Grant 511568 with the acronym 3DTV. It is also partially supported by the Academy of Finland, project No. 213462 (Finnish Centre of Excellence program (2006 - 2011)) and The Scientific and Technological Research Council of Turkey (TUBITAK). *Bunny* model is courtesy of

Stanford and *Venus head* model is courtesy of Cyberware. We would like to thank Andrei Khodakovsky for PGC software and Alexander Bogomjakov for vertex ordering software in Ref.[13]

REFERENCES

1. P. Alliez and C. Gotsman, "Recent advances in compression of 3D meshes," in *Proc. Symposium on Multiresolution in Geometric Modeling*, 2003.
2. J. Peng, C.-S. Kim, and C.-C. J. Kuo, "Technologies for 3D mesh compression: A survey," *Journal of Visual Communication and Image Representation* **16**, pp. 688–733, Dec. 2005.
3. Z. Yan, S. Kumar, and C.-C. J. Kuo, "Error resilient coding of 3-D graphic models via adaptive mesh segmentation," *IEEE Trans. Circuits Syst. Video Technol.* **11**, pp. 860–873, Jul. 2001.
4. G. AlRegib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively transmitted 3-D models," *IEEE Trans. Multimedia* **7**, pp. 766–776, Aug. 2005.
5. V. Goyal, "Multiple description coding: compression meets the network," *IEEE Signal Processing Mag.* **18**, pp. 74–93, September 2001.
6. P. Jaromersky, X. Wu, Y. Chiang, and N. Memon, "Multiple-description geometry compression for networked interactive 3D graphics," in *Proc. ICIG'2004*, pp. 468–471, Dec. 2004.
7. M. O. Bici and G. Bozdagi Akar, "Multiple description scalar quantization based 3D mesh coding," in *Proc. IEEE Int. Conf. Image Processing*, (Atlanta, US), Oct. 2006.
8. A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," in *Siggraph 2000, Computer Graphics Proceedings*, pp. 271–278, 2000.
9. C. Touma and C. Gotsman, "Triangle mesh compression," in *Proc. Graphics Interface*, (Vancouver, BC, Canada), Jun. 1998.
10. A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.* **6**, pp. 243–250, June 1996.
11. A. Miguel, A. Mohr, and E. Riskin, "SPIHT for generalized multiple description coding," in *Proc. IEEE Int. Conf. Image Processing*, **3**, pp. 842 – 846, Oct. 1999.
12. W. Jiang and A. Ortega, "Multiple description coding via polyphase transform and selective quantization," in *Proc. Visual Commun. Image Processing (VCIP'99)*, pp. 53–63, (San Jose,CA), Jan. 2000.
13. A. Bogomjakov and C. Gotsman, "Universal rendering sequences for transparent vertex caching of progressive meshes," *Computer Graphics Forum* **21**, pp. 137–148, 2002.
14. Z. Karni, A. Bogomjakov, and C. Gotsman, "Efficient compression and rendering of multi-resolution meshes," in *Proc. IEEE Visualization*, (Boston, US), Oct. 2002.
15. E. Riskin, "Optimum bit allocation via generalized BFOS algorithm," *IEEE Trans. Inform. Theory* **37**, pp. 400–4002, March 1991.
16. P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," *Computer Graphics Forum* **17**, pp. 167–174, 1998.
17. Y. Charfi, R. Hamzaoui, and D. Saupe, "Model-based real-time progressive transmission of images over noisy channel," in *Proc. WCNC'03*, pp. 347–354, (New Orleans, LA), Mar. 2003.