compressed 3D meshes

Shakeel Ahmad, Raouf Hamzaoui, Marwan Al-Akaidi

Abstract

We consider a state of the art system that uses layered source coding and forward error correction with Reed-Solomon codes to efficiently transmit 3D meshes over lossy packet networks. Given a transmission bit budget, the performance of this system can be optimized by determining how many layers should be sent, how each layer should be packetized, and how many parity bits should be allocated to each layer such that the expected distortion at the receiver is minimum. The previous solution for this optimization problem uses exhaustive search, which is not feasible when the transmission bit budget is large. We propose instead an exact algorithm that solves this optimization problem in linear time and space. We illustrate the advantages of our approach by providing experimental results for the CPM (Compressed Progressive Meshes) mesh compression technique.

I. Introduction

An increasing number of applications require fast delivery of three-dimensional (3D) models over an unreliable channel. For example, Internet users may want to browse a graphic library consisting of many 3D objects, which are typically modeled as 3D triangular meshes. To enable high-quality renderings, millions of triangles may be needed for a single object. Thus, in addition to a fast and reliable error control technique, a compression scheme that provides a compact representation of the 3D mesh is crucial. While 3D mesh compression is a mature research area, only a few 3D mesh error control techniques have been proposed. Yan, Kumar, and Kuo [1], [2] partition a 3D mesh into small pieces that are compressed independently to avoid error propagation and use error concealment techniques at the receiver to recover the missing data. A similar idea was proposed by Bischoff and Kobbelt [3]. Chen, Bodenheimer, and Barnes [4] use a progressive compression scheme that generates a base mesh and a number of refinement layers. The base mesh and a portion of the refinement layers are sent with the transmission control protocol (TCP), while the remaining layers are sent without protection with the user datagram protocol (UDP). In [5], the same authors propose a different approach based on UDP and forward error correction. Bici, Norkin, and Akar [6] use the wavelet-based Progressive Geometry Compression scheme [7] to compress the 3D mesh

This paper appears in IEEE Transactions on Multimedia, vol. 11, pp. 1381–1387, Nov. 2009. The paper was presented in part at ICME-06, IEEE International Conference on Multimedia and Expo, Toronto, July 2006. The authors are with the Faculty of Technology, De Montfort University, The Gateway, Leicester LE1 9BH, UK. Email: sahmad@dmu.ac.uk, rhamzaoui@dmu.ac.uk, mma@dmu.ac.uk. Tel: Shakeel Ahmad 44 116 207 8973, Raouf Hamzaoui 44 116 207 8096, Marwan Al-Akaidi 44 116 257 7087.

and the forward error correction system of [8] to protect the resulting embedded bitstream. The wavelet bitstream is packetized into a block of packets where unequal error protection is applied across the packets. However, the framework of [8] does not allow progressive transmission. Also it cannot be used with large 3D models because of practical limits on the packet size and channel codeword length [9].

In this paper, we focus on a system proposed by AlRegib, Altunbasak, and Rossignac [10], which also relies entirely on forward error correction (see also [11] and [12] for variations of this system and [13] for an extension that exploits selective retransmission). But unlike the system used in [6], it allows progressive transmission and can be used with large models. A progressive compression scheme is used to generate a base mesh and a number of refinement layers. Any transmitted layer is protected by applying Reed-Solomon codes across the packets and packetized in a block of packets (BOP) consisting of information symbols and parity symbols. Given a total transmission bit budget, the performance of the system is optimized by finding the number of layers that should be transmitted and the number of parity symbols allocated to each transmitted layer such that the expected distortion at the receiver is minimum. To solve this optimization problem, AlRegib *et al.* [10] apply exhaustive search whose complexity is reduced by constraining the number of parity symbols assigned to a given layer to be greater than or equal to the number of parity symbols assigned to the next layer. We show that even with this monotonicity constraint, the time complexity of exhaustive search remains too high, making it unfeasible when large models are used. We also show that this constraint can lead to poor performance and propose a dynamic programming algorithm that solves the unconstrained optimization problem in linear time and space.

Our algorithm is inspired from an algorithm of Dumitrescu and Wu [14], which was developed to optimize a related system [15]. However, the algorithm of Dumitrescu and Wu [14] cannot be used with the system of [10]. Indeed, the packet size is fixed a priori in the system of [15], while it can vary with the BOP in the system of [10]. Second, the number of information symbols allocated to a BOP is variable in the system of [15], whereas it is fixed in the system of [10]. Third, the number of transmitted BOPs is fixed in the system of [15], whereas it must be optimized in [10]. We also point out that our algorithm has linear time and space complexity, while the algorithm of [14] has quadratic time and space complexity.

The paper is organized as follows. In Section II, we discuss the system proposed in [10]. We show in particular that the proposed heuristic algorithm to optimize the bit allocation problem for this system is unfeasible when the 3D model is large. In Section III, we propose a better packetization technique for this system and give an algorithm that finds an optimal solution to the bit allocation problem in linear time and space. In Section IV, we give experimental results for 3D models compressed with the CPM coder [16]. We consider both a two-state Markov channel and a simulated Internet link.

II. Previous work

Consider media data compressed to M layers consisting of one base layer and $M-1$ update layers. To decode a given layer, all information symbols in this layer and in the previous layers must be available. Let s_j denote the

i	i	i	i	i	i	i
i	i	i	i	i	i	i
x	x	x	x	i	i	x
x	x	x	x	x	x	x
x	x	x	x	x	x	x

TABLE I

$L = 3$ LAYERS ARE SENT IN THREE BOPS OF $N = 5$ PACKETS EACH. THE LENGTH OF THE PACKETS IS $m_1 = 4$, $m_2 = 2$, AND $m_3 = 1$ IN THE FIRST, SECOND, AND THIRD BOP, RESPECTIVELY. INFORMATION SYMBOLS ARE DENOTED BY I AND PARITY SYMBOLS BY X.

number of symbols (for example, bytes) in the jth layer ($j = 1, \ldots, M$). For $j = 1, \ldots, L$ ($1 \leq L \leq M$), the jth layer is packetized into a group of k_j horizontal information packets of size m_j symbols each. Then the same systematic Reed-Solomon code (or punctured or shortened Reed-Solomon code) of length N is applied vertically on each of the m_j groups of k_j information symbols, yielding a BOP consisting of N channel packets of size m_j symbols each. The number N of channel packets in a BOP is fixed, while both k_j and m_j are variable and satisfy $1 \leq k_j \leq N$, $k_j m_j = s_j$, and $(N - k_j)m_j = c_j$, where c_j is the number of parity symbols assigned to the jth layer (Table I).

Because Reed-Solomon codes are maximum distance separable codes, when layer j is transmitted over a packet erasure channel, the receiver can recover all s_j information symbols if no more than $N - k_j$ channel packets are lost out of the N transmitted ones.

Suppose now that the total transmission budget is fixed to $S = s_1 + \cdots + s_M$. Then one can either send all layers without protection, or the first $M-1$ layers with a total protection budget $c_1 + \cdots + c_{M-1} = s_M$, or the first $M-2$ layers with a total protection budget $c_1 + \cdots + c_{M-2} = s_M + s_{M-1}$, etc. That is, the total transmission budget is kept constant by trading off information symbols for parity symbols. When L BOPs are sent $(1 \leq L \leq M)$ and an *error protection allocation* $\mathbf{c}_L = (c_1, \ldots, c_L)$ is used, the expected distortion is

$$E(\mathbf{c}_L) = (1 - B_1(c_1))E_0 + \sum_{j=2}^{L} E_{j-1}(1 - B_j(c_j)) \prod_{k=1}^{j-1} B_k(c_k) + E_L \prod_{k=1}^{L} B_k(c_k) \tag{1}$$

where $B_j(c)$ is the probability of successfully recovering all the information symbols in the jth layer when $c_j = c$, and E_j is the distortion if the first j layers are reconstructed. Here we assume that packet losses are independent in different BOPs. Note that E_j is the error between the reconstructed 3D model if all M layers are decoded and the reconstructed 3D model if only the first j layers are decoded. The goal is to determine the number of transmitted layers L and the associated error protection allocation $\mathbf{c}_L = (c_1, \ldots, c_L)$ such that the expected distortion (1) is minimum. AlRegib *et al.* [10] compute a solution to this problem by full search. This is done as follows. For all values of L, $L = M, \ldots, 1$, with corresponding protection budget C_L, $C_L = 0, s_M, s_M + s_{M-1}, \ldots, s_M + s_{M-1} + \cdots + s_2$, all possible allocations $\mathbf{c}_L = (c_1, \ldots, c_L)$ such that $c_1 + \cdots + c_L = C_L$ are tested, and the best one is selected. To reduce the complexity of this exhaustive search, they propose to allow only solutions that satisfy the constraint $c_1 \geq \ldots \geq c_L$. Moreover, they suggest to stop the iteration on L in the algorithm as soon as the expected distortion of the best allocation for L is greater than that for the previous L.

The approach of [10] has a major drawback. When C_L is large, the number of possible L-tuple candidates $(c_1, \ldots, c_L)$ that satisfy the constraint $c_1 + \cdots + c_L = C_L$ and $c_1 \geq \ldots \geq c_L$ is too large to allow an exhaustive search. For example, when $L = 11$, the number of candidates is 42, 560, 20298, 2012069, and 470259534 for $C_L = 10, 20, 40, 80,$ and 160, respectively.

Because of the BOP packetization constraints $1 \leq k_j \leq N, k_j m_j = s_j$, and $(N - k_j)m_j = c_j$, not all candidates are admissible. But this is not an advantage as usually only a few candidates will be admissible, leading to a poor performance of the system. Worse, in many cases, there will be no admissible allocations. For example, when $s_1 = 4269$, $N = 100$, and $C_L = 2650$, it is not possible to build a BOP for the first layer.

III. Proposed solution

In this section, we first present a flexible BOP packetization technique. Then we provide a linear-time algorithm that finds an optimal number of layers to be transmitted and a corresponding optimal error protection allocation. In contrast to [10], we do not assume that the number of parity symbols is nonincreasing. Moreover, we do not assume that the transmission budget is fixed to $s_1 + \cdots + s_M$. Instead, we compute our solution for any arbitrary total transmission budget.

A. *Proposed packetization*

We define the code gain g of a BOP as the number of packets that contain only parity symbols. By increasing the code gain of a BOP, we make it more robust against packet loss, but we increase the total number of transmitted symbols for this BOP.

With the packetization of [10], the code gain g may not take all values in the set $\{0, \ldots, N-1\}$. For example, suppose that $N = 5$ and $s_j = 34$. Then only $g = 3$ and $g = 4$ are admissible. We propose to make the construction of a BOP more flexible by inserting a dummy information symbol at the end of an information packet if needed. In this way, we guarantee that g can take all values in $\{0, \ldots, N-1\}$. We do not transmit the dummy symbols as they can be inserted at the decoder automatically just to realize a rectangular BOP and hence do not contribute to the transmitted bit budget. For the example with $N = 5$ and $s_j = 34$, the values $g = 0, 1, 2$ can be obtained with $(m_j = 7, d_j = 1), (m_j = 9, d_j = 2)$, and $(m_j = 12, d_j = 2)$, respectively (Table II). Here d_j denotes the total number of dummy symbols for the BOP. The N possible packetizations corresponding to the N values $g = 0, \ldots, N-1$ can be obtained in a systematic way as follows. For a given code gain $g \in \{0, \ldots, N-1\}$, let $m_{j,g}, d_{j,g}$, and $k_{j,g}$ denote the size of the packet, the number of dummy symbols, and the number of information packets, respectively. Then we have $k_{j,g} = N - g$ and $(m_{j,g}, d_{j,g})$ is the unique pair solving the equation $k_{j,g}m_{j,g} = s_j + d_{j,g}$ subject to $0 \leq d_{j,g} \leq k_{j,g} - 1$. The number of parity symbols associated to a code gain g for the layer j is $c_{j,g} = gm_{j,g}$.

i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	i
i	i	i	i	i	i	d

i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	d
i	i	i	i	i	i	i	i	d
x	x	x	x	x	x	x	x	x

i	i	i	i	i	i	i	i	i	i	i	i
i	i	i	i	i	i	i	i	i	i	i	d
i	i	i	i	i	i	i	i	i	i	i	d
x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x

TABLE II

THREE OF THE FIVE POSSIBLE PACKETIZATIONS OF LAYER j INTO A BOP WHEN $N = 5$ AND $s_j = 34$. THE LETTER D DENOTES A DUMMY SYMBOL.

B. Proposed algorithm

Using the BOP packetization of Section III-A, we provide an algorithm for finding an optimal number of transmitted layers $L \in \{1, \ldots, M\}$ and a corresponding optimal code gain allocation $\mathbf{g}_L = (g_1, \ldots, g_L)$ given a total transmission budget R. Note that once an optimal code gain allocation is known, one can derive immediately the corresponding optimal error protection allocation $\mathbf{c}_L = (c_{1,g_1}, \ldots, c_{L,g_L})$ and packets sizes $m_{1,g_1}, \ldots, m_{L,g_L}$ as explained in Section III-A.

The bit allocation problem is illustrated in Fig. 1. Each node at stage i gives an available transmission budget for sending layers $i, \ldots, M$. As we move down from a node at stage i to a node at stage $i+1$, the available transmission budget is decreased by the number of source symbols at layer i plus the number of parity symbols used to protect layer i with the code gain corresponding to the edge connecting the two nodes. A node at stage i is connected to a node at stage $i+1$ if the available transmission budget at node i is enough to send layer i with the code gain shown by the edge between the two nodes. Thus we might have leaves at different levels. Each path connecting the root node to a leaf is associated with an expected distortion. We have to choose a path that gives the smallest expected distortion. The length of the chosen path gives an optimal number L of layers to be transmitted and the edges along this path give the corresponding optimal code gains.

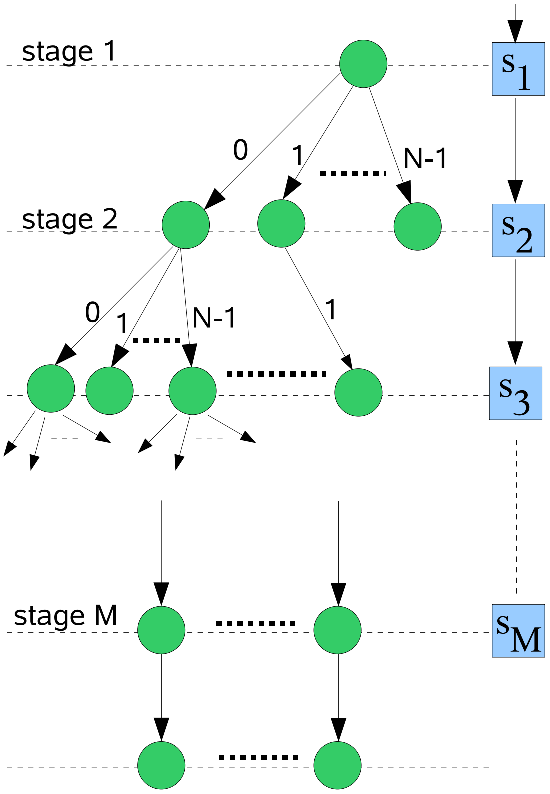


Fig. 1. A solution to the optimization problem is given by a path from the root to a leaf with minimum expected distortion. The length of the chosen path gives an optimal number of layers to be transmitted. Optimal code gains for each layer to be transmitted are indicated by the edges along the selected path. An optimal path can be found recursively starting from bottom to top.

We observe that for a given node, the optimal sub-path below the node is independent of the sub-path above the node. This leads to the idea of solving the problem recursively by traversing the tree from bottom to top. Below we prove that our observation is true and give a formal description of the algorithm.

Let $P(x) = \sum_{j=0}^{x} p_N(j)$, where $p_N(j)$ is the probability of losing j packets out of N transmitted ones. Let $\Delta E(g_i, \ldots, g_L)$ denote the expected reduction in the distortion when layers $i, \ldots, L$ are sent with code gains $g_i, \ldots, g_L$, respectively, given that layers $1, \ldots, i-1$ are correctly decoded. Then

$$\Delta E(g_i, \ldots, g_L) = \sum_{t=i}^{L} (E_{t-1} - E_t) \prod_{j=i}^{t} P(g_j).$$

Since the expected error at the receiver is

$$E(g_1, \ldots, g_L) = E_0 - \Delta E(g_1, \ldots, g_L)$$

an optimal code gain allocation should maximize the expression $\Delta E(g_1, \ldots, g_L)$. This can be done recursively with dynamic programming using the following result.

Lemma 1: For all $k = i+1, \ldots, L$

$$\Delta E(g_i, \ldots, g_L) \;=\; \Delta E(g_i, \ldots, g_{k-1}) + \Delta E(g_k, \ldots, g_L) \prod_{x=i}^{k-1} P(g_x).$$

Proof. Let $\Delta E_t(g) = (E_{t-1} - E_t)P(g)$. Then

$$\begin{aligned}\Delta E(g_i, \ldots, g_L) &= \sum_{t=i}^{L} \Delta E_t(g_t) \prod_{j=i}^{t-1} P(g_j) \\ &= \sum_{t=i}^{k-1} \Delta E_t(g_t) \prod_{j=i}^{t-1} P(g_j) + \prod_{x=i}^{k-1} P(g_x) \sum_{t=k}^{L} \Delta E_t(g_t) \prod_{j=k}^{t-1} P(g_j)\end{aligned}$$

which gives the desired result.

Let $P_i(r, g)$ be equal to $P(g)$ if $r \geq s_i + c_{i,g}$ and equal to zero, otherwise. Let $\Delta E_i(r, g)$ be equal to $\Delta E_i(g)$ if $r \geq s_i + c_{i,g}$ and equal to zero, otherwise. Let $\Delta G(r; i : L)$ denote the maximum expected reduction in distortion when layers $i, \ldots, L$ are sent over the channel with code gains $g_i, \ldots, g_L$, respectively, given that $\sum_{j=i}^{L} s_j + c_{j,g_j} \leq r$, and layers $1, \ldots, i-1$ were correctly decoded. That is,

$$\Delta G(r; i : L) = \max_{\sum_{j=i}^{L} s_j + c_{j,g_j} \leq r} \Delta E(g_i, \ldots, g_L).$$

Then Lemma 1 with $k = i + 1$ gives for $1 \leq i < L$

$$\Delta G(r; i : L) = \max_{0 \leq g \leq N-1} \{\Delta E_i(r, g) + P_i(r, g)\Delta G(r - s_i - c_{i,g}; i + 1 : L)\} \qquad (2)$$

On the other hand, we have

$$\Delta G(r; L : L) = \max_{0 \leq g \leq N-1} \Delta E_L(r, g). \qquad (3)$$

In a preprocessing step, we compute for each layer $i = 1, \ldots, M$ and for each code gain $g = 0, \ldots, N-1$, the set of parity symbols $c_{i,g}$ (see Section III-A). Then we build the $M \times (R+1)$ array $\Delta G(r; i : M)$ for $i = M, \ldots, 1$ and $r = 0, \ldots, R$, recursively using (2) and (3).

Now, let $g^*(\Delta G(r; i : L))$ be a function that returns the smallest g that achieves $\Delta G(r; i : L)$ in (2) (respectively (3)) when $\Delta G(r; i : L) > 0$ and let it be 0 otherwise. Then $g^*(\Delta G(r; i : L))$ is the optimal code gain g for the ith layer (and the corresponding $c_{i,g}$ is the optimal number of parity symbols) given that r is the number of available transmission symbols for layers i to L, and that layers 1 to $i-1$ are already decoded.

Algorithm 1 summarizes our approach. The time complexity of the algorithm is $O(NMR)$, and its space complexity is $O(MR)$. One can speed up Algorithm 1 by noting that for each i, the array entry $\Delta G(r; i : M)$ need only be computed for $r = s_i$ to $R - \sum_{j=1}^{i-1} s_j$ as all other array entries are either 0 or will never be required.

Note that the algorithm can also be used when the size of a channel packet is constrained to be smaller than a given maximum length or larger than a given minimum length. In this situation, we simply include the constraint in the preprocessing step.

Algorithm 1 Optimal solution

```
**Input**: $M$ (number of layers), $s_i,\ i = 1, \dots, M$ (number of symbols in layer $i$), $E_i,\ i = 0, \dots, M$ (distortion when $i$ layers are decoded), $N$ (number of packets in a BOP), $p_N(i),\ i = 0, \dots, N$ (probability of losing $i$ packets out of $N$), $R$ (transmission budget).
**Output**: $L$ (number of layers to be transmitted), $g_i,\ i = 1, \dots, g_L$ (code gain for layer $i$).
for $i = M$ to $1$ do
    for $r = 0$ to $R$ do
        Compute $\Delta G(r; i : M)$ and $g^*(\Delta G(r; i : M))$
    end for
end for
Set $r = R$ and $L = 0$.
for $i = 1$ to $M$ do
    if $r \geq s_i$ then
        $g_i = g^*(\Delta G(r; i : M))$
        $r = r - (s_i + c_{i,g_i})$
        $L = i$
    else
        break
    end if
end for
```

IV. Experimental results

We provide experimental results for the transmission of 3D models with the system described in Section III. We compare the optimal solution (S1) obtained by Algorithm 1, a solution (S2) that assumes as in [10] that the number of parity symbols assigned to a layer is nonincreasing ($c_{i+1,g_{i+1}} \leq c_{i,g_i}$), and a solution (S3) that assumes that the code gain is nonincreasing ($g_{i+1} \leq g_i$). Note that S2 and S3 cannot be computed by exhaustive search even for modest-sized instances of the problem. In [17], we give algorithms that compute S2 and S3 in $O(NMR)$ and $O(MR^2)$, respectively. However, these algorithms cannot be used with large models because their space complexity is too large. For this reason, we first present simulation results for a small Bunny model (Fig. 2).

We compressed the model with the CPM coder [16] to produce one base mesh and 11 update layers. The base mesh was further compressed with the Edge Breaker algorithm [18]. The number of information symbols

Fig. 2. Bunny model consisting of 5,597 triangles and 2,907 vertices.

(here bytes) was $s_1 = 4269, s_2 = 117, s_3 = 144, s_4 = 185, s_5 = 240, s_6 = 293, s_7 = 415, s_8 = 624, s_9 = 892, s_{10} = 1284, s_{11} = 1805$, and $s_{12} = 2650$. The distortions measured by the quadric error metric [19] were $E_1 = 1695, E_2 = 973.64, E_3 = 721.47, E_4 = 292.18, E_5 = 202.30, E_6 = 138.18, E_7 = 84.63, E_8 = 42.08, E_9 = 21.54, E_{10} = 11.33, E_{11} = 4.76$, and $E_{12} = 0$. The parameter E_0 corresponds to the distortion when no layers are received. The choice of this parameter has an effect on the solution computed by the algorithms. The greater the value of E_0, the more importance is given to the protection of the base layer. In this experiment, we set $E_0 = 5000$.

Fig. 3 gives the expected distortion (1) as a function of the mean packet loss rate for the three solutions. The channel model was a two-state Markov process. This channel model can be characterized by the transitional probabilities from the bad state to the good state or by the mean packet loss rate and the average loss burst length (that is, the average number of consecutively lost packets) [20], [10]. The number of channel packets per BOP was set to $N = 100$. The total transmission budget was $R = \sum_{j=1}^{12} s_j = 12918$ bytes. Tables III and IV show the allocation of the parity symbols and the assignment of the code gains to the transmitted layers for mean packet loss rate 0.2 (only eight layers were transmitted at this packet loss rate).

The experiments show that the monotonicity constraint on the number of protection symbols can lead to a significant loss in performance. Fig. 4 shows that this result also holds for the root mean square error (computed with the Metro tool [21]). Here $E_1 = 41.43, E_2 = 32.51, E_3 = 22.70, E_4 = 16.81, E_5 = 11.449, E_6 = 8.91, E_7 = 6.53, E_8 = 4.51, E_9 = 2.84, E_{10} = 1.84, E_{11} = 1.03, E_{12} = 0.0$, and $E_0 = 186.56$ was the root mean square error between the original model and its bounding box.

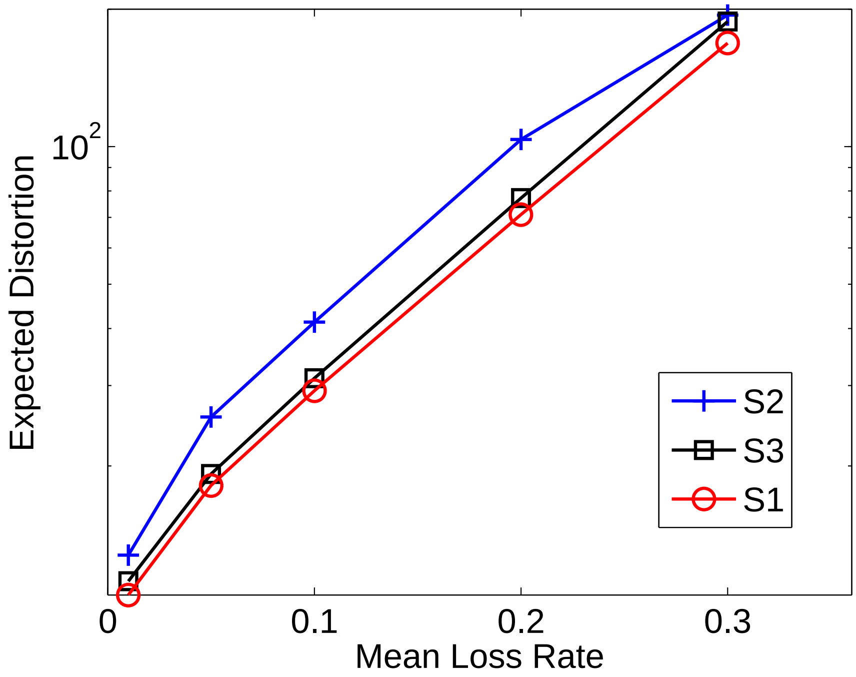


Fig. 3. Expected distortion (quadric metric) as a function of the channel mean packet loss rate for the Bunny model. The channel is modeled as a two-state Markov process with burst length 5. S1 is the optimal solution, S2 assumes nonincreasing number of parity symbols, and S3 assumes nonincreasing code gains.

Layer	1	2	3	4	5	6	7	8
S1	4628	183	240	290	260	300	360	370
S2	4628	280	256	212	210	205	180	176
S3	4823	159	156	208	260	294	360	370

TABLE III

NUMBER OF PARITY BYTES ALLOCATED TO THE TRANSMITTED LAYERS BY S1, S2, AND S3. RESULTS CORRESPOND TO FIG. 3 WITH MEAN PACKET LOSS RATE 0.2.

Although the individual symbols in layer i are more important than those in layer $i+1$, the protection of layer i may require fewer parity symbols if this layer has fewer information symbols. The experiments also show that the monotonicity constraint on the code gains can harm the performance of the system.

The expression for the expected distortion (1) assumes that the packet losses are independent across different BOPs. This assumption is not true in channels with memory. However, Fig. 5 shows that when the burst length is not too large, this assumption does not harm the performance of the system. Fig. 6 and 7 confirm this result for a Dragon (Fig. 8) and Budha (Fig. 9) models progressively compressed to 21 and 12 layers, respectively. Since the layers have now a much larger size, the value of N was increased to 200. Moreover, to obtain realistic packet sizes, Algorithm 1 was run under minimum and maximum packet size constraints of 100 and 1460 bytes, as explained

Layer	1	2	3	4	5	6	7	8
S1	52	61	60	58	52	50	45	37
S2	52	70	64	53	42	41	30	22
S3	53	53	52	52	52	49	45	37

TABLE IV

Code gains assigned to the transmitted layers by S1, S2, and S3. Results correspond to Fig. 3 with mean packet loss rate 0.2.

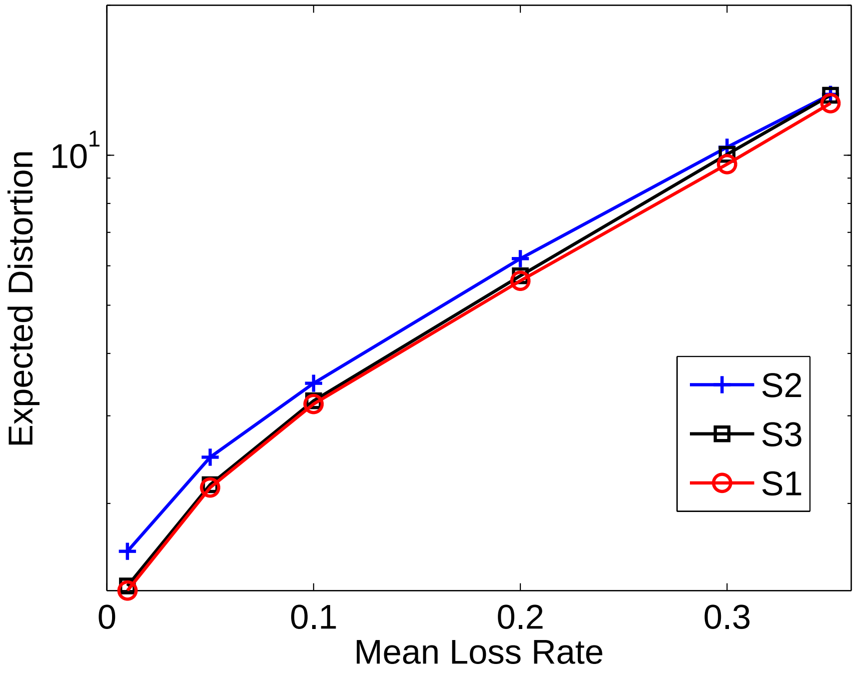


Fig. 4. Expected distortion (root mean square error) as a function of the channel mean packet loss rate for the Bunny model. The channel is modeled as a two-state Markov process with burst length 5. S1 is the optimal solution, S2 assumes nonincreasing number of parity symbols, and S3 assumes nonincreasing code gains.

in Section III-B.

In the last experiment, we studied the performance of the optimal solution (S1) on a simulated Internet link. We collected a one-hour packet loss trace by sending ICMP packets of size 500 bytes from a computer in Konstanz, Germany, to a remote machine in Minsk, Belarus, which reflected all received packets back to Konstanz. We modeled the Konstanz-Minsk-Konstanz link as a two-state Markov channel and used the collected trace to estimate the mean packet loss rate and the average burst error length. We then applied our algorithm to determine the optimal error protection for the Budha model for the estimated channel parameters (0.0467 for the mean packet loss rate and 3.2 for the average burst length) and various transmission bit budgets. The expected distortion matched the actual average distortion (Fig. 10). Fig. 11 shows the progressive average reconstruction quality at the receiver side

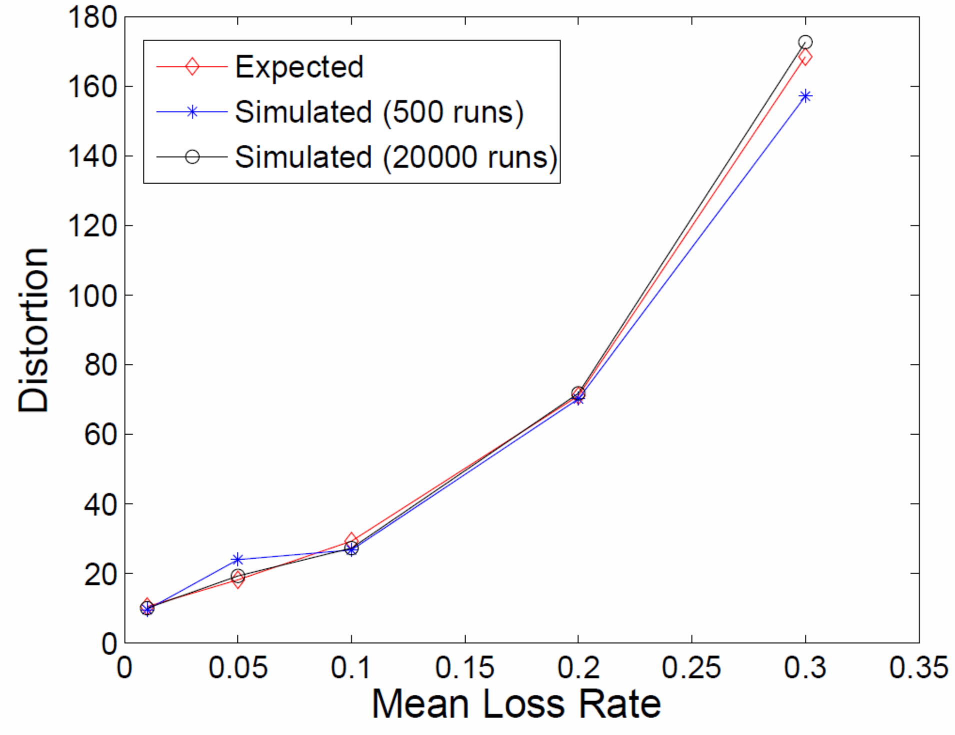


Fig. 5. Expected and average distortion (quadric error) resulting from 500, and 20000 simulation runs. The same settings as in Fig. 3 are used.

when the total number of transmitted bytes is $820,416$. The pictures show the layer corresponding to the nearest receiver average distortion.

V. Conclusion

We proposed a flexible packetization technique and an efficient source-channel allocation algorithm for the state of the art transmission system of AlRegib *et al.* [10]. Although our algorithm has linear-time complexity, its CPU time is not small enough for applications where the solution has to be computed in real time. However, since our algorithm finds an optimal solution, it can be used to check the quality of suboptimal solutions computed with heuristic techniques. It can also be used to build a lookup table that gives the solutions for a large set of channel conditions. Finally, we point out that although the system of AlRegib *et al.* [10] was originally introduced for the progressive transmission of 3D models, it can also be used with any kind of progressively compressed media data, including JPEG2000 bitstreams.

Acknowledgment. We thank Renato Pajarola for providing us with the source code of the CPM algorithm. This work was supported by the DFG Research Training Group GK-1042.

References

[1] 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.*, vol. 11, pp. 860–873, Jul. 2001.

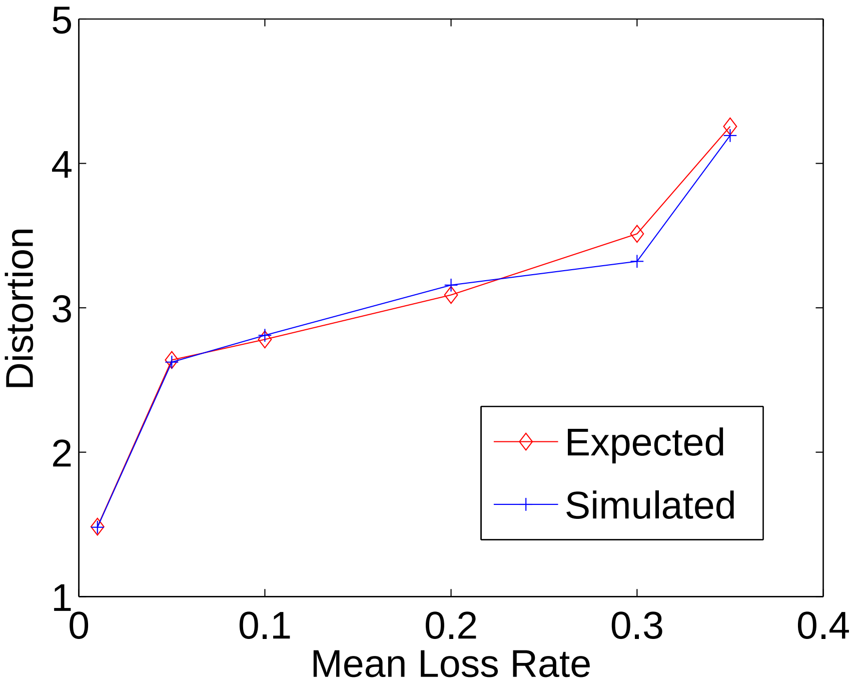


Fig. 6. Expected and average distortion (quadric error) resulting from 4000 simulation runs for the Dragon model. The total transmission bit budget is $1,616,922$ bytes.

[2] Z. Yan, S. Kumar, and C.-C. J. Kuo, "Mesh segmentation schemes for error resilient coding of 3-D graphic models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, pp. 138–144, Jan. 2005.

[3] S. Bischoff and L. Kobbelt, "Streaming 3D geometry data over lossy communication channels," *Proc. IEEE ICME 2002*, Lausanne, Aug. 2002.

[4] Z. Chen, B. Bodenheimer, and J. F. Barnes, "Robust transmission of 3D Geometry over lossy networks," *Proc. Int. Conf. on 3D Web Technology 2003*, pp. 161-173, St. Malo, France, 2003.

[5] Z. Chen, J. F. Barnes, and B. Bodenheimer, "Hybrid and forward error correction transmission techniques for unreliable transport of 3D geometry," *Multimedia Systems* , vol. 10, no. 3, pp. 230–244, March, 2005.

[6] M.O. Bici, A. Norkin, G.B. Akar, "Packet loss resilient transmission of 3D models," *Proc. IEEE ICIP 07*, vol. 5, pp. 121–124, San Antonio, TX, Sept. 2007.

[7] A. Khodakovsky, P. Schröder, and W. Sweldens, "Progressive geometry compression," *Proc. ACM Siggraph*, pp. 271–278, 2000.

[8] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 819-828, Dec. 2000.

[9] R. Hamzaoui, V. Stanković, and Z. Xiong, "Optimized error protection of scalable image bitstreams," *IEEE Signal Proc. Mag.*, vol. 22, pp. 91–107, Nov. 2005.

[10] G. AlRegib, Y. Altunbasak, and J. Rossignac, "Error-resilient transmission of 3-D models," *ACM Trans. Graphics*, pp. 182–208, April 2005.

[11] G. AlRegib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively transmitted 3-D models," *IEEE Trans. Multimedia*, vol. 7, pp. 766–776, Aug. 2005.

[12] G. AlRegib, Y. Altunbasak, and R. Mersereau, "Bit allocation for joint source and channel coding of progressively compressed 3D meshes," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 15, pp. 256–268, Feb. 2005.

[13] D. Tian and G. AlRegib, "On-demand transmission of 3D models over lossy networks,". *Signal Processing: Image Communication*,

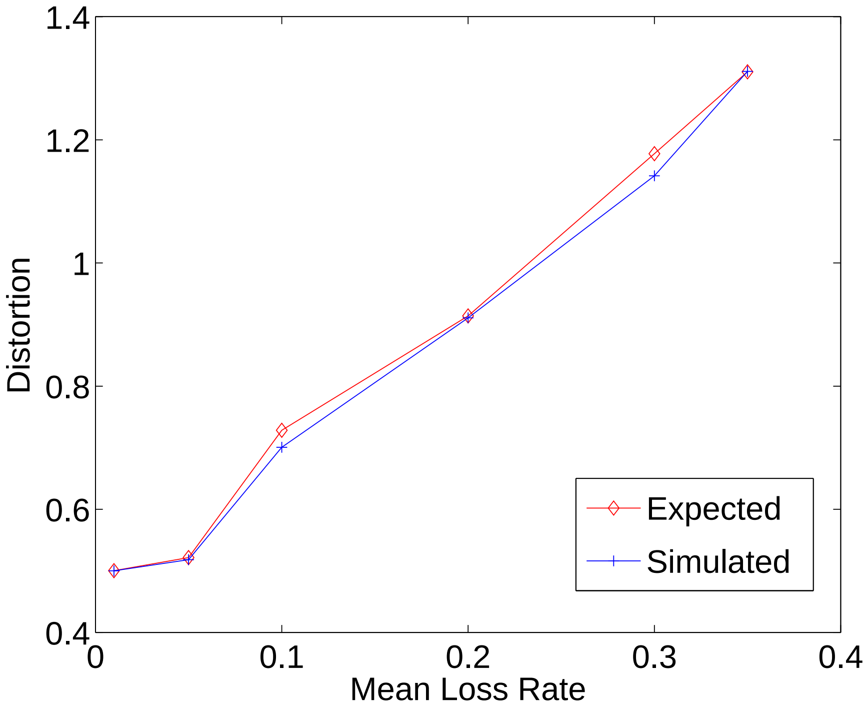


Fig. 7. Expected and average distortion (root mean square error) resulting from 4000 simulation runs for the Budha model. The total transmission bit budget is $820,416$ bytes.

vol. 21, pp. 396–415, June 2006.

[14] S. Dumitrescu and X. Wu, “Globally optimal uneven erasure-protected multi-group packetization of scalable codes,” *Proc. IEEE ICME 2005*, pp. 900–903, Amsterdam, July 2005.

[15] J. Thie and D. Taubman, “Optimal erasure protection assignment for scalable compressed data with small channel packets and short channel codewords,” *EURASIP Journal on Applied Signal Processing*, no. 2, pp. 207–219, Feb. 2004.

[16] R. Pajarola and J. Rossignac, “Compressed progressive meshes,” *IEEE Trans. Visual. Comput. Graphics*, vol. 6, no. 1, pp. 79–93, 2000.

[17] S. Ahmad and R. Hamzaoui, “Optimal error protection of progressively compressed 3D meshes”, Konstanzer Schriften in Mathematik und Informatik, Technical Report 222, University of Konstanz, `http://www.ub.uni-konstanz.de/kops/volltexte/2007/2277/`.

[18] H. Lopes, J. Rossignac, A. Safonova, A. Szymczak, and G. Tavares, “Edgebreaker: A simple compression algorithm for surfaces with handles,” *Computers & Graphics*, vol. 27, no. 4, pp. 553–567, 2003.

[19] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” *Proc. SIGGRAPH 97*, pp. 209–216, 1997.

[20] U. Horn, K. Stuhlmller, M. Link, and B. Girod, “Robust internet video transmission based on scalable coding and unequal error protection,” *Signal Processing: Image Communication*, vol. 15, no. 1–2, pp. 77–94, Sept. 1999.

[21] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: measuring error on simplified surfaces, *Computer Graphics Forum*, vol. 17, pp. 167-174, 1998.

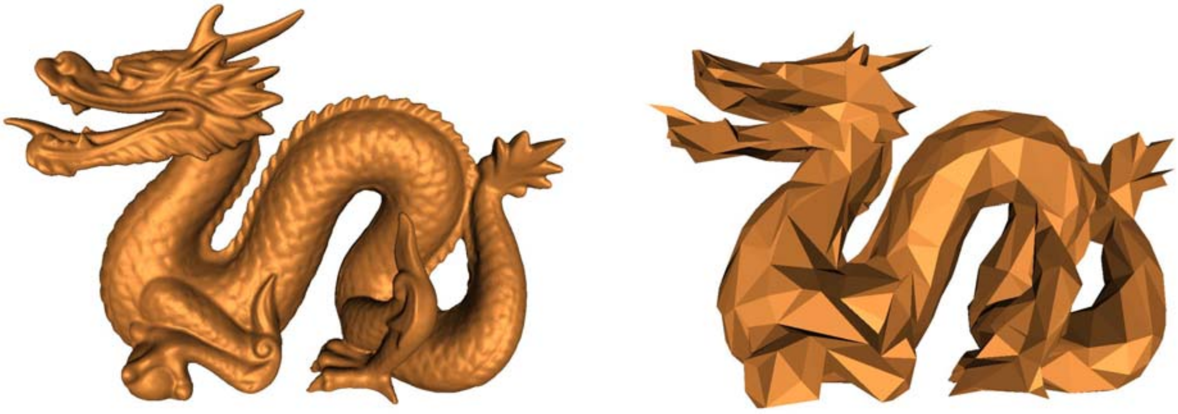


Fig. 8. Dragon model consisting of 871,414 triangles and 437,645 vertices. (left) Original model. (right) Base layer.

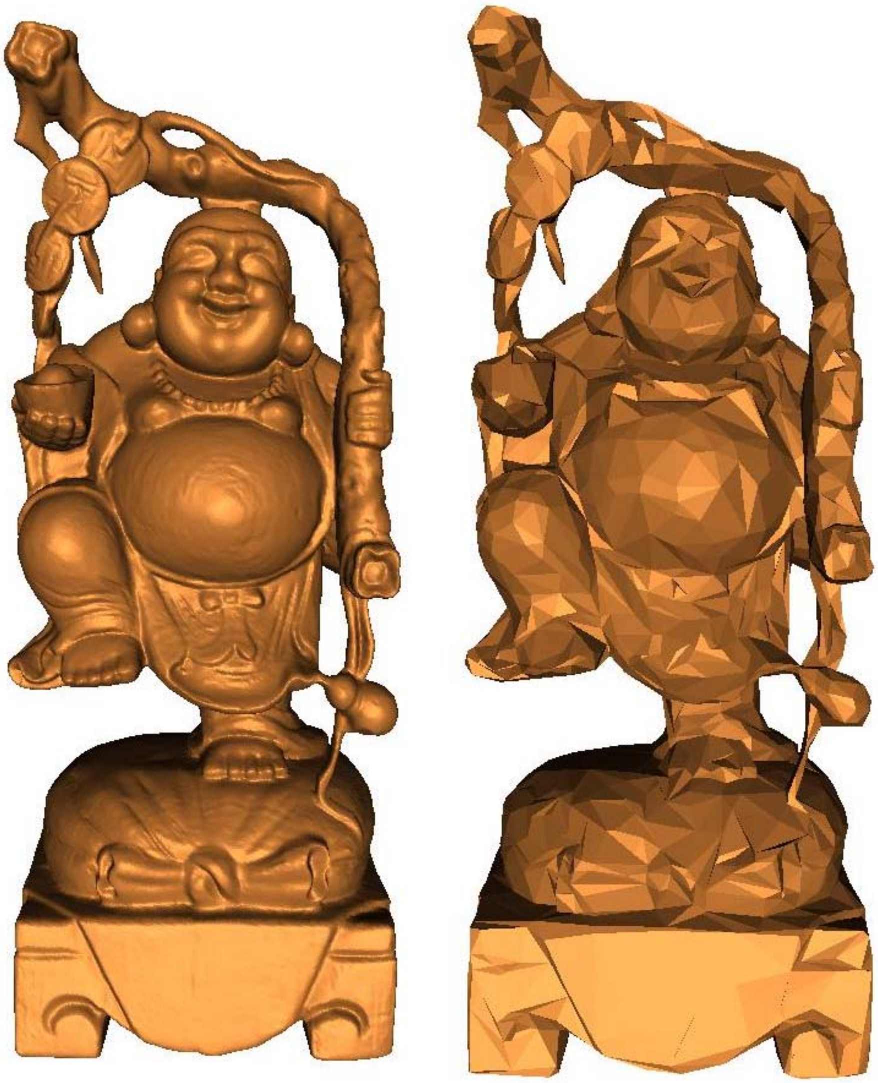


Fig. 9. Budha model consisting of 499,996 triangles and 249,988 vertices. (left) Original model (right) Base layer.

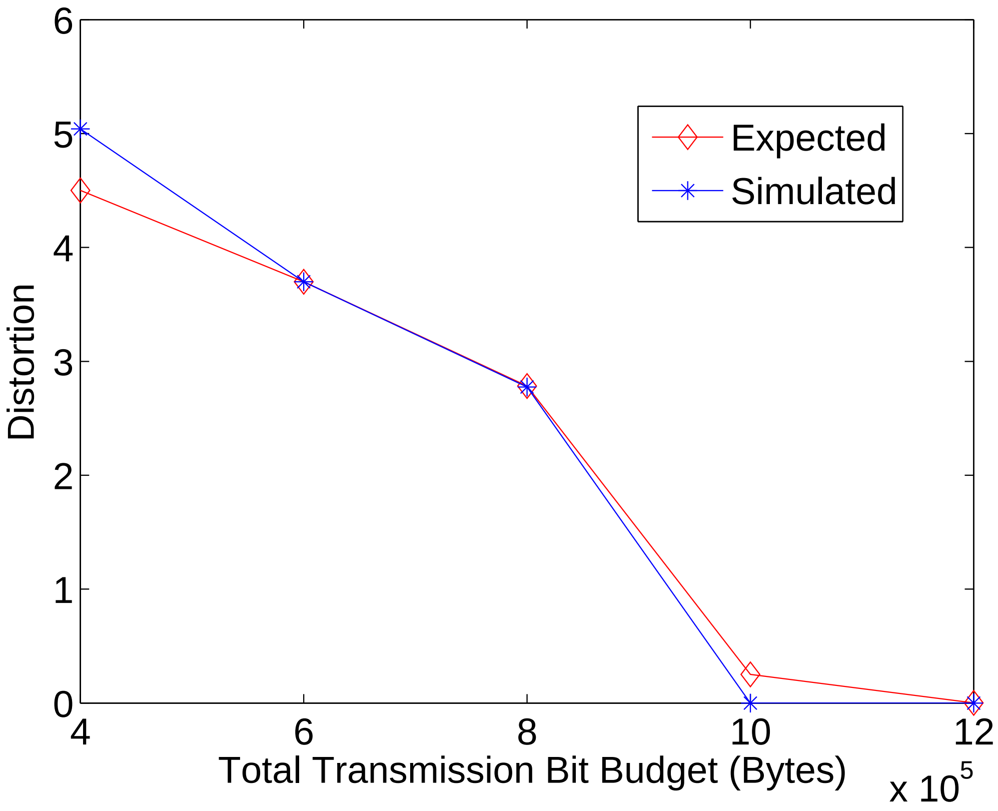
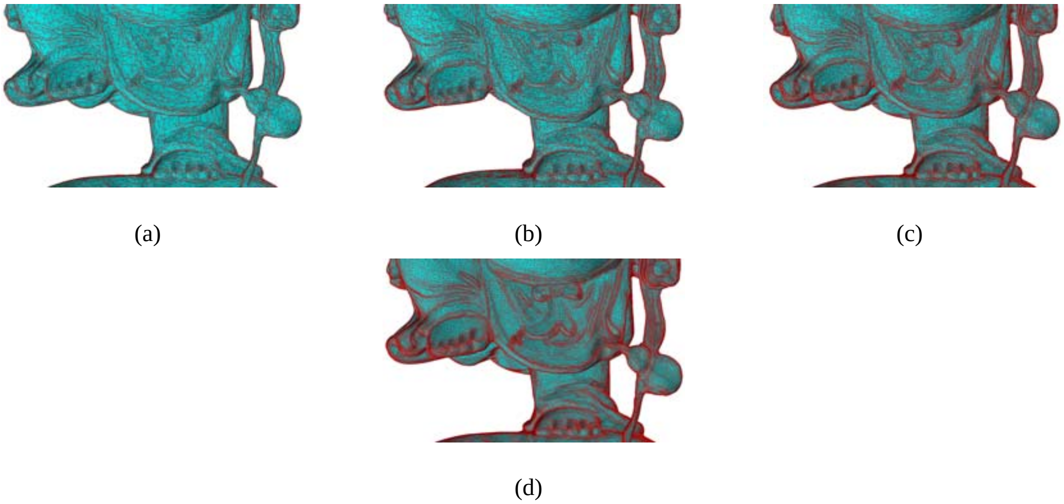


Fig. 10. Expected and average distortion (quadric error) as a function of the total transmission bit budget for the Budha model on the Internet link Konstanz-Minsk-Konstanz. The average distortion is computed for 10,000 transmissions of the model.


(a)

(b)

(c)

(d)

Fig. 11. Average reconstruction quality for the Budha model after (a) 25 % (b) 50 % (c) 75 % (d) 100 % bits of a total $820,416$ bytes are sent.