Available online at www.sciencedirect.com

SCIENCE DIRECT®

Signal Processing: *Image Communication* 21 (2006) 396–415

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.com/locate/image

On-demand transmission of 3D models over lossy networks

Dihong Tian, Ghassan AlRegib*

School of Electrical and Computer Engineering, Georgia Institute of Technology, 210 Technology Circle, Savannah, GA 31407-3039, USA

Received 7 September 2005; accepted 25 January 2006

Abstract

Three-dimensional (3D) meshes are used intensively in distributed graphics applications where model data are transmitted on demand to users' terminals and rendered for interactive manipulation. For real-time rendering and high-resolution visualization, the transmission system should adapt to both data properties and transport link characteristics while providing scalability to accommodate terminals with disparate rendering capabilities. This paper presents a transmission system using hybrid unequal-error-protection and selective-retransmission for 3D meshes which are encoded with multi-resolutions. Based on the distortion-rate performance of the 3D data, the end-to-end channel statistics and the network parameters, transmission policies that maximize the service quality for a client-specific constraint is determined with linear computation complexity. A TCP-friendly protocol is utilized to further provide performance stability over time as well as bandwidth fairness for parallel flows in the network. Simulation results show the efficacy of the proposed transmission system in reducing transmission latency and providing smooth performance for interactive applications. For example, for a fixed rendering quality, the proposed system achieves 20–30% reduction in transmission latency compared to the system based on 3TP, which is a recently presented 3D application protocol using hybrid TCP and UDP.

Keywords: Distributed 3D graphics; Interaction; Multi-resolution compression; Unequal error protection; Selective retransmission; TCP-friendly rate control; Multimedia streaming

1. Introduction

Internet-based multimedia applications are expanding from streaming video/audio to distributed 3D graphics, driven by growing demands of various applications such as electronic commerce, collaborative CAD, medical and scientific visualization, and virtual environments. Interaction is one of the key aspects of a distributed graphics application. Most of the distributed graphics applications require certain level of interaction with the objects involved. Some applications such as immersive environments and shared reality thoroughly involve users in the interaction. Response time, which is the latency between the user input and the response (e.g., scenes displayed on the user's terminal) from the system, is one of the major considerations in designing high-performance distributed graphics systems. In contrast to specific 3D systems that assume all models are locally available and are essentially designed as stand-alone systems, distributed graphics applications often require on-demand exchange of 3D data in a networked environment, and impose requirements of real-time response and

*Corresponding author. Tel.: +1 912 966 7937; fax: +1 912 966 7928.

E-mail addresses: dhtian@ece.gatech.edu (D. Tian), gregib@ece.gatech.edu (G. AlRegib).

0923-5965/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2006.01.003

smooth performance over time on the interaction. In addition, 3D data differ from general media content for it requires rendering capability on the user terminal. Sending the same dimension of data to user terminals with disparate rendering capabilities may result in significant difference in response time between the different clients. Scalability of the 3D data, as well as the transmission, is therefore desirable. All the aforementioned requirements, not only have promoted the use of high-performance computing systems and distributed platforms, but also call for careful considerations on ways of reducing transmission latency, providing scalability, and maintaining high-resolution visualization when network delays and random data losses are involved.

Multi-resolution compression of 3D meshes [1,16,19–21,23,24] is a partial solution to provide scalability for 3D data. Using multi-resolution encoders, the server can select the appropriate resolution for a particular client according to its quality requirement, or initially sends a coarse representation of the 3D model to the client for quick reconstruction and rendering, and then transmits refinement layers which allow the client to gradually increase model fidelity toward higher resolutions. Although, such methods are successful in exploring the space and time efficiency of 3D data, higher efficiency can be accomplished by addressing the effects of network behaviors. In particular, to display 3D scenes on the user's terminal with satisfactory quality and in real time, the impact of packet losses and transmission delays on the decoding process need to be explored. Typically, reliable or error-resilient transmission can be achieved by pre-processing techniques such as data partitioning [7,17,27], post-processing techniques such as error-concealment, and network-oriented techniques such as forward error correction [2,6,4] and retransmission techniques [3,5,8,9,14]. All these techniques address efficient transmission of 3D data *separately*. Yet an appropriate combination of such techniques is desired to achieve better performance. Interaction and tradeoffs among the selected techniques need to be investigated, taking into account the distortion-rate performance of the 3D data and the network characteristics.

In this paper, a hybrid mechanism of unequal-error-protection and selective-retransmission is proposed for multi-resolution meshes. Hierarchical data batches of the multi-resolution mesh are protected preferentially according to their distortion-rate performance, network parameters, and channel statistics estimated by the transport layer. To minimize response time in interaction, the proposed mechanism is designed to have linear computational complexity. In addition, by integrating TCP-friendly congestion control into the system, the proposed mechanism achieves smooth performance over time as well as bandwidth fairness for co-existing applications in the network. Simulation results show the efficacy of the proposed mechanism. For instance, compared with a recently presented 3D application protocol referred to as 3TP [5], the proposed system achieves 20–30% reduction in transmission latency while delivering the same level of rendering quality.

The main contributions of the presented work are summarized as follows:

- We analyze intensively the property of multi-resolution 3D meshes and present a TCP-friendly transmission system for multi-resolution meshes including a novel and meaningful quality measure.
- Given a distortion constraint, we derive a simplified expression (with assumptions) of the optimal FEC code that minimizes the expected transmission latency when combined with retransmissions.
- Observing a semi-infinite space for finding the theoretical optimal solution, we propose an extended steepest decent search algorithm which quickly reaches the local optima in the solution space.
- Based on the above results for quality-critical scenarios, we further develop a time-critical streaming algorithm which significantly decreases the receiving distortion upon a strict delay constraint.

The rest of the paper is organized as follows. The relevant prior art that addresses channel effects in multi-resolution mesh transmission is summarized in the next section. Major aspects of the proposed mesh transmission system are described in Section 3, and Section 4 presents a detailed study of the hybrid unequal-error-protection and selective-retransmission mechanism. Test results in simulated network environments are given in Section 4. Finally, Section 5 concludes the paper and summarizes future work.

2. Prior art

In this section, we briefly discuss transmission systems that have been proposed in the literature [2–9,14,17,27]. According to the adopted error-resilient mechanism, these systems can be categorized into pre-processing, error correction, and transport-layer protocols. In MPEG-4, pre-processing-based error resiliency is achieved by data partitioning [7,17,27], i.e., partitioning the bitstream into segments and decoding each segment independently. In [14], the partitions are ordered in a tree structure according to their interdependencies and transmitted accordingly.

Using forward error correction (FEC) in providing error resiliency to multi-resolution 3D meshes has been investigated by AlRegib et al. [2,4,6], where unequal-error-protection (UEP) for different resolutions of the compressed mesh is proposed. In [4,6], bit-allocation algorithms are presented to distribute source and channel coding bits under a total bit budget. The similar concept is also exploited in this work for forward packet-loss resilience, but our work differs from those earlier approaches in an essential aspect: instead of solely using FEC, the proposed solution *jointly* considers forward packet-loss resilience and feedback-based retransmission. As a result, the system supports reliable mesh transmission and can guarantee a certain quality level required by the application. Moreover, the proposed mechanism determines the optimal tradeoff between forward packet-loss resiliency and retransmission such that the delay is minimal with respect to the distortion constraint.

The transport control protocol (TCP) and the user datagram protocol (UDP) are usually used for error-sensitive and delay-sensitive streams, respectively. Recently, hybrid TCP/UDP transport protocols such as 3TP have been proposed to stream 3D models [3,5,9]. The general idea is to send important data reliably using TCP and the remaining, less important, data using UDP. In spite of the effectiveness of these algorithms, they do not fully explore the space or time efficiency of the lossy channel as they utilize solely feedback-based retransmission without considering its interaction with error control coding. In addition, rate control is not considered when using UDP, which makes the application irresponsive to congestion and unfairness to other streams in the network. The lack of a rate control mechanism also results in large variation of the transmission throughput, and therefore, of the receiving quality over time. Finally, an essential difficulty of using hybrid TCP and UDP is the synchronization between the two transport associations. In contrast to the hybrid TCP/UDP transport protocols, the transmission mechanism proposed in this paper employs a unified transport layer, and regulates data retransmission at the application level. Transmission latency is further reduced by incorporating unequal error protection with retransmission. A TCP-friendly rate control protocol is implemented in the presented transmission system, which provides both smoothness and responsiveness to the application.

3. System overview

In this section, we provide an overview on the 3D mesh transmission system that is under consideration, and introduce the most important system parameters. As can be seen in Fig. 1, the proposed system has three major components: a 3D mesh codec, a hybrid unequal-error-protection and selective-retransmission mechanism at the application level (UEP/SR), and a transport protocol integrated with TCP-friendly rate control (TFRC). Next, we briefly describe the system components and the corresponding parameters. Toward the end of the section, we present a simulation environment which will be used to test the performance of the proposed system.

3.1. The codec component

The 3D mesh codec implemented in the present work closely follows the *compressed progressive mesh* algorithm proposed by Pajarola and Rossignac [21] although the proposed transmission system is general enough to incorporate other multi-resolution compression methods, as will be discussed later in this section. A compressed mesh stream is composed of a base mesh, M_0, and L enhancement batches, $\{B_i\}_{i=1,\ldots,L}$, each of which encodes a set of *vertex-split* [16] operations which transfer the triangulated mesh surface to a higher resolution.[1] Sequentially, batch B_i refines the

[1] For those methods that do not require grouping data into batches to achieve high compression ratio such as a wavelet-based scheme [19], we still consider this "batched" organization as a general representation of multi-resolution mesh data for two reasons: (i) isolated refinement operations in the multi-resolution mesh perform small and localized changes which normally do not result in perceivable distortion, and (ii) the refinement data will

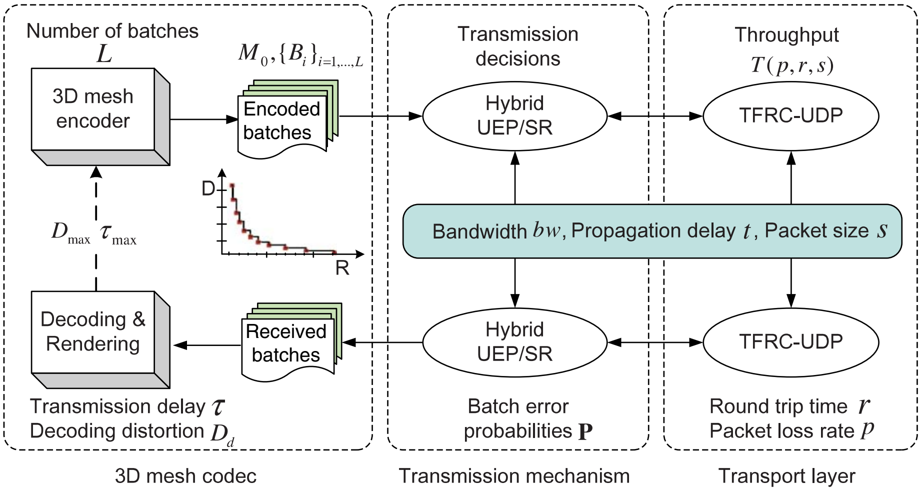

Fig. 1. The proposed transmission system consists of three components.

resolution of mesh M_{i-1} to higher resolution M_i until the full resolution is reached. Each resolution M_i, differs from the full resolution mesh by certain error (distortion), D_i, with a certain bit-rate, R_i. A measure of such distortion that properly reflects the perceptual quality of different resolutions is important. In this paper, we measure such distortion by introducing a metric with a similar formulation to the peak signal-to-noise ratio (PSNR) which is commonly used in imaging. In particular, for a mesh with multiple resolutions $\{M_i\}_{i=0,\ldots,L}$ where M_0 is the base mesh and M_{L} is the full resolution, we define the quality of mesh M_i as

$$\mathrm{PSNR}_m \triangleq 20\log_{10}\frac{\mathscr{E}_{\max}(M_0, M_{\mathrm{L}})}{\mathscr{E}_{\mathrm{rms}}(M_i, M_{\mathrm{L}})} \quad (\mathrm{dB}), \qquad (1)$$

where $\mathscr{E}_{\max}()$ and $\mathscr{E}_{\mathrm{rms}}()$ are the measured maximum and root-mean-square surface distances between the corresponding pairs of meshes, respectively. Details on this metric are given in Appendix A. These distances can be calculated using the fast Metro tool [10] in practice. In addition, an empirical study presented in [26] and our further discussion in Appendix A show that using this PSNR-like metric has the following advantages: (i) it provides meaningful measurement on visual quality of different resolutions, (ii) it normalizes the distance and provides unified quantities for describing quality of multiple models that are in the same 3D scene but with different coordinate grids, and (iii) the numerical range of the metric is close to that of PSNR in imaging, which brings convenience for evaluating the quality of 3D models based on human's subjective experience. These aspects are generally important for quality control in graphics applications and are especially vital for applications that involve users in distributed and interactive 3D presentations.

For a compressed multi-resolution mesh, transformation from a coarser representation to a finer resolution includes decoding two parts of information: connectivity and geometry. Connectivity information encodes the *cut-edges* for performing vertex-split operations, which refine the topology of the mesh surface on the contrary of edge-collapses [16]; geometric data record the position of the collapsed vertex, and is compressed using vertex prediction followed by entropy coding for the prediction error [21]. If the connectivity is not decodable because of information loss, the transformation will stop because the vertex-split process cannot continue without knowing the cut-edges. On the other hand, it is not the case when any geometry information is missing because geometry contains only prediction errors of vertex positions, which do not prevent the decoding process from proceeding to the next level but introduce additional distortion. It should be mentioned that in a wavelet-based mesh compression scheme such as [18,19], all connectivity information except for the coarsest representation is eliminated by converting the input mesh to a semi-regular mesh before compression, in which case only

(*footnote continued*)
eventually be packetized when they are transmitted over the network.

vertex-position data are encoded in the enhancement bitstream.

For the codec using edge-collapse and vertex-split as basic operations, both connectivity and geometry are included in enhancement batches. Intuitively, one may consider that connectivity information is more important than geometry and should be treated differently in transmission. However, our empirical study (Appendix B) shows that, because of error propagation, decoding connectivity without refining geometry tends to amplify the geometric error and results in degradation instead of improvement on visual quality. In other words, although connectivity and geometry have different impacts on the decoding process for the multi-resolution mesh, they are equally important in the sense of preserving visual quality. Therefore, while unequal error resilience is desired for different data batches from an optimal quality point of view regarding their unequal rate-distortion performance, information encoded within a batch should be treated in a unified manner. Importantly, this fact provides generality to the proposed system to incorporate all 3D model codecs that are either based on edge-collapse/vertex-split operations or wavelets.

In all present multi-resolution mesh coding schemes, the base mesh M_0 has both connectivity and geometry information and has to be correctly received or the rendering process will not be able to start (*infinite* distortion), while it in general has a fairly small fraction (1–2% or less) of the entire bitstream. Regarding this, to simplify notation (avoid differentiating the base mesh and the enhancement batches), and for the ease of performance presentation (avoid presenting infinite distortion), in the rest of the paper we assume that a coarse representation has been received by the client through a reliable channel, and focus our discussion on the transmission of enhancement data.

3.2. The transmission component

The L enhancement batches, $\{B_i\}_{i=1,\ldots,L}$, have different rate-distortion performance and are treated intelligently to achieve the best quality. In this paper, best quality is interpreted as either minimized transmission latency τ under a distortion constraint $D_{\max}$ or minimized decoding distortion D_{d} within a limited time frame $\tau_{\max}$. Given network statistics reported by the transport layer, the sending application protects the batches with unequal-rate forward error correction (FEC) codes and/or retransmission according to their respective costs, and determines the optimal tradeoff that minimizes the transmission delay τ under the constraint of $D_{\max}$ or vice versa. This hybrid unequal-error-protection and selective retransmission mechanism (UEP/SR) is the core contribution of this work. To illustrate, consider the situation where a distortion constraint $D_{\max}$ is imposed. A set of χ batches, $\{B_1, B_2, \ldots, B_\chi\}$, are selected under a criterion that their overall bit-rate is minimal while their resulting resolution has a distortion level below the constraint, i.e., $D_{\mathrm{d}} \leqslant D_{\max}$. According to the discussion in Section 3.1, these selected batches need to be reliably transmitted, or the decoding distortion of the received mesh will not be satisfied (if any batches among the selected ones are not correctly received, sending higher batches would not help decreasing the decoding distortion). Moreover, instead of solely using feedback-based retransmission to achieve reliable transmission, unequal-rate FEC codes are concurrently used to reduce the potential retransmission cost, and computation is performed to find the optimal FEC codes that minimize the expected transmission delay, $E(\tau)$, with the condition that undecodable batches among the selected ones will be retransmitted. Detailed discussion on the transmission mechanism is presented in Section 4.

The Reed–Solomon (RS) code is employed for FEC. We assume an (n, k) RS code with a block size of n packets (i.e., $n \times$ *packet-size* symbols) including k $(k < n)$ information packets (code rate k/n). Considering a lossy channel, an RS code with $(n-k)$ parity-check packets will be able to recover the same number of packet losses. Hence the error probability of a batch using RS code (n, k) is

$$P(n,k) = Pr\{>(n-k) \text{ losses out of } n\}. \tag{2}$$

If the random packet loss is modeled by a Bernoulli process, for example, it is easy to get

$$\begin{aligned} P(n,k) &= 1 - Pr\{\leqslant (n-k) \text{ losses out of } n\} \\ &= 1 - \sum_{i=0}^{n-k} \binom{n}{i} p^i (1-p)^{n-i}, \end{aligned} \tag{3}$$

where p is the packet-loss rate. Similarly, the calculation of (2) can be performed for more sophisticated channel models, for which a few results have been found in the literature. For a two-state Markov channel model, for example, one may refer to the derivation presented in [15]. As will be shown in Section 4, such results can be easily integrated in the computation of transmission decisions, and the simulation results show that even the simple approximation (3) performs fairly well with the proposed algorithms.

3.3. The transport layer

UDP streams suffer from the lack of congestion control mechanism that prevents them from being reasonably fair[2] when competing for bandwidth with TCP-based traffic, as TCP throttles its transmission rate against the network congestion. A TCP-friendly system should regulate its data sending rate according to the network condition, typically expressed in terms of the packet size s, the round-trip time r, and the packet-loss rate p. Ideally, the TCP throughput equation is suitable in describing the steady-state sending rate of a TCP-friendly flow [11]:

$$T = \frac{s}{r\sqrt{2p/3} + 4r(3\sqrt{3p/8})p(1+32p^2)}, \tag{4}$$

where a recommended choice of the retransmission timeout, $t_{\mathrm{RTO}} = 4r$, has already been integrated.

A TFRC protocol based on Eq. (4) has been specified in [13]. TFRC is a receiver-based mechanism designed for applications that use a fixed packet size and vary their sending rate in packets per second in response to congestion. The receiver periodically (once per round-trip time) sends feedback reports to the sender, containing the information that allows the sender to adjust its sending rate. Rate control using TFRC provides bandwidth fairness for parallel flows in the network as well as network stability, thus avoiding congestion collapse. In addition, TFRC's rate fluctuation is much lower than TCP, making it more appropriate for streaming applications that desire constant receiving quality. The TFRC implemented in the proposed mesh transmission system is slightly modified so that each data packet is acknowledged by the receiver [22]. The ACKs are used to infer the round-trip time and detect packet losses in order to perform selective retransmission for undecodable batches.

3.4. Simulation environment

In this section, we present a simulation environment on which all our reported results will be based.

(1) *Network environment*: Simulation in this paper is performed using ns-2 [25]. Fig. 2 shows the simulated topology. The link between $R1$ and $R2$ is the bottleneck and the bandwidth is shared by f parallel flows. Random early detection (RED) [12] gateways are deployed at $R1$ and $R2$ to improve both fairness and performance of the flows. All the parameters are listed in Table 1.

(2) *Test models*: Simulation results, unless otherwise noted, are obtained using the following models (courtesy of Cyberware, Inc): Horse (39 698 faces), Dinosaur (28 096 faces), Venus Head (67 170 faces), and Balljoint (68 530 faces). All the models are quantized with 12 bits[3] and are encoded to generate 10 enhancement batches. Their rate-distortion performance is plotted in Fig. 3.

4. The proposed transmission mechanism

In a hybrid TCP/UDP protocol as reviewed in Section 2, transmission latency is reduced by using UDP while distortion is reduced by TCP, i.e., feedback-based retransmission. Error control coding is an alternative to retransmission to improve the channel utilization and provide quality control. Although information theory has shown that sole use of feedback or error control coding can achieve the channel capacity, real systems often have constraints that invalidate ideal assumptions and call for joint considerations on both approaches. Typically, there are many situations where the receiving application has a maximally allowed level on either the transmission latency or the rendering distortion. In such cases, retransmission and error control coding should be treated intelligently so that the lowest distortion level or the minimum transmission latency is obtained while satisfying the corresponding constraint. In this section, we study in detail the hybrid unequal-error-protection and selective-retransmission mechanism for multi-resolution meshes. This proposed mesh transmission

[2] A flow is "reasonably fair" if its sending rate is generally within a factor of two of the sending rate of a TCP flow under the same condition [13].

[3] In general, using 12–14 bits for quantizing 3D models with moderate dimensions introduces invisible distortion [21]. It is worthwhile to be pointed out, however, that the test models and the quantization parameter herein are selected for the illustration purpose. Our proposed transmission mechanism has no dependency on these parameters.

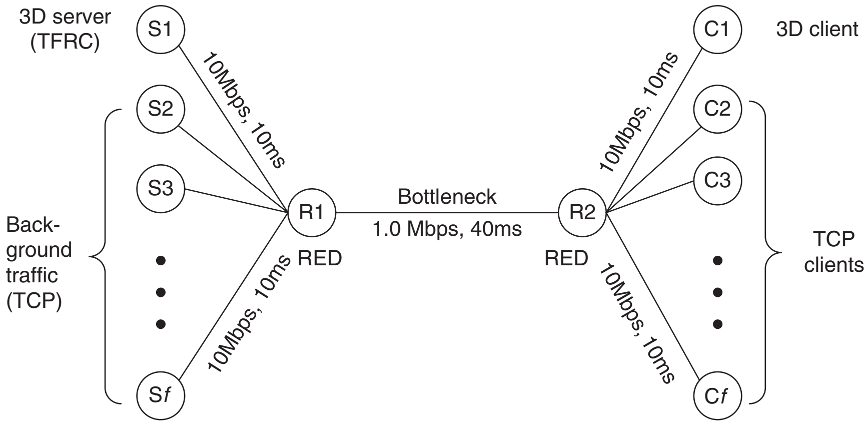

Fig. 2. Simulation topology in ns-2. There are totally *f* parallel flows sharing the bottleneck bandwidth. All the background traffic is FTP traffic transmitted over TCP.

Table 1
Summary of simulation parameters

Packet size	1000/500 bytes
ACK size	40 bytes
Bottleneck delay	40 ms
Bottleneck bandwidth	1 Mbps
Bandwidth of side links	10 Mbps
Delay of side links	10 ms
Simulation duration	100–200 s
TCP window	20
RED parameters	
Min threshold	5 packets
Max threshold	0.5*buffer size
Buffer size	1000 packets
*q*_weight	0.002

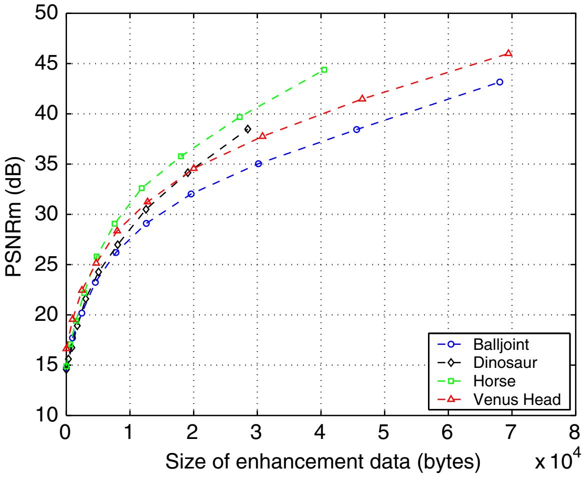

Fig. 3. Rate-distortion performance for four test models.

mechanism is referred to as REP (retransmission and error protection) hereafter.

4.1. *Distortion-constrained transmission*

We first look at the problem of minimizing the transmission latency with a given distortion constraint, $D_{\max}$. For a multi-resolution mesh stream, if a lower batch is not decoded correctly, perceptual quality cannot be improved by decoding higher batches. Henceforth, satisfying $D_{\max}$ is equivalent to selecting the least number of batches that need to be transmitted reliably. Denoted by χ, this least number of batches is expressed as

$$\chi = \min\{x|D_{\mathrm{d}}(x) \leqslant D_{\max}\}, \tag{5}$$

where $D_{\mathrm{d}}(x)$ denotes the decoding distortion of the first x batches.

Once the number of selected batches is determined, the challenge is to find the optimal tradeoff between forward error protection and retransmission such that the user-requested distortion level can be satisfied with minimum transmission latency. To do so, an optimal distribution of parity-check packets for the selected batches is computed, taking into account their unequal rate-distortion performance and potential retransmission costs. The selected data are then protected with the corresponding parity-check packets, and is transmitted (with possible retransmissions) until the batches are correctly decoded. A simple illustration of this transmission mechanism is presented in Fig. 4, where χ batches are selected to be reliably transmitted with a calculated distribution of the

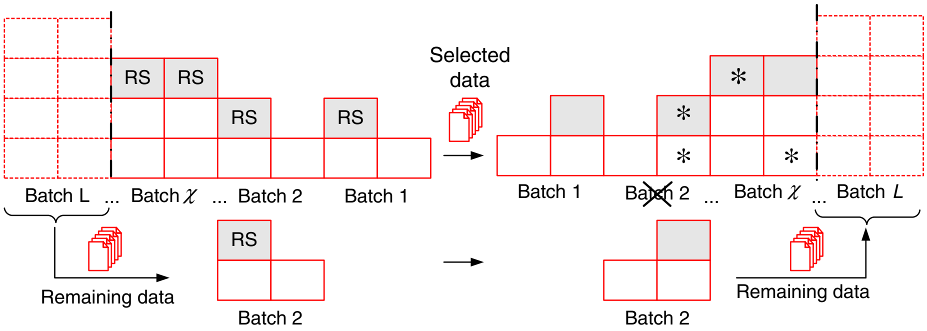
RS
RS
RS
RS
RS
Selected data
*
*
*
*
Batch L ... Batch χ ... Batch 2 Batch 1
Batch 1 Batch 2 ... Batch χ ... Batch L
Remaining data
Batch 2
Remaining data
Batch 2

Fig. 4. A simple demonstration on the proposed hybrid UEP/SR method, where the packets marked with 'RS' are parity-check packets and '*' indicates packet losses during transmission. Batch 2 is retransmitted because it is among the selected χ batches and has not been correctly received.

parity-check packets (noted with 'RS'). Packets marked with '$*$' indicate the losses occurred during transmission. As depicted in Fig. 4, Batch 2 is not decodable due to packet losses and is retransmitted in order to satisfy the distortion constraint. After the χ batches are correctly received, the remaining data are transmitted (with or without error resilience, provided that the requested distortion level has been met).

Using $\mathbf{s}_\chi$ and $\mathbf{c}_\chi$ to represent the vectors (with length χ) of source and parity-check packets, respectively, for the selected χ batches, the expectation of the transmission latency τ is given as

$$\begin{aligned} E(\tau|\mathbf{s}_\chi,\mathbf{c}_\chi) &= E(\tau_{\mathbf{s}_\chi}+\tau_{\mathbf{c}_\chi}+\tau_R|\mathbf{s}_\chi,\mathbf{c}_\chi) \\ &= E(\tau_{\mathbf{s}_\chi}+\tau_{\mathbf{c}_\chi}|\mathbf{s}_\chi,\mathbf{c}_\chi)+E(\tau_R|\mathbf{s}_\chi,\mathbf{c}_\chi), \end{aligned} \tag{6}$$

where $E()$ represents the probabilistic expectation, and τ_R denotes the total latency incurred by retransmission; $\tau_{\mathbf{s}_\chi}$, $\tau_{\mathbf{c}_\chi}$ denote the transmission costs for all the source and parity-check packets, respectively, in the selected batches, i.e.,

$$\tau_{\mathbf{s}_\chi}=|\boldsymbol{\tau}_{\mathbf{s}_\chi}|=\sum_{i=1}^{\chi}\boldsymbol{\tau}_{\mathbf{s}_\chi}(i) \quad\text{and}\quad \tau_{\mathbf{c}_\chi}=|\boldsymbol{\tau}_{\mathbf{c}_\chi}|=\sum_{i=1}^{\chi}\boldsymbol{\tau}_{\mathbf{c}_\chi}(i),$$

where $\boldsymbol{\tau}_{\mathbf{s}_\chi}(i)$ and $\boldsymbol{\tau}_{\mathbf{c}_\chi}(i)$ correspond to the transmission cost for the ith batch. Given the RS codes $(\mathbf{s}_\chi,\mathbf{c}_\chi)$, the packet-loss rate p and the round-trip time r, the steady-state transmission throughput T is described by Eq. (4), and $\boldsymbol{\tau}_{\mathbf{s}_\chi}$, $\boldsymbol{\tau}_{\mathbf{c}_\chi}$ can be considered as constant-value vectors. With the notations $\boldsymbol{\tau}_0=(\boldsymbol{\tau}_{\mathbf{s}_\chi}+\boldsymbol{\tau}_{\mathbf{c}_\chi})$ and $\tau_0=\tau_{\mathbf{s}_\chi}+\tau_{\mathbf{c}_\chi}=|\boldsymbol{\tau}_0|=\mathbf{1}\cdot\boldsymbol{\tau}_0$, we have $E(\tau_{\mathbf{s}_\chi}+\tau_{\mathbf{c}_\chi}|\mathbf{s}_\chi,\mathbf{c}_\chi)=E(\tau_0)$, and Eq. (6) is further expressed as

$$\begin{aligned} E(\tau|\mathbf{s}_\chi,\mathbf{c}_\chi) &= E(\tau_0)+E(\tau_R|\mathbf{s}_\chi,\mathbf{c}_\chi) \\ &= |\boldsymbol{\tau}_0|+\mathbf{P}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)\cdot[\boldsymbol{\tau}_0+E(\boldsymbol{\tau}_R|\mathbf{s}_\chi,\mathbf{c}_\chi)] \\ &= \mathbf{1}\cdot\boldsymbol{\tau}_0+\mathbf{P}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)\cdot\boldsymbol{\tau}_0 \\ &\quad +\mathbf{P}^2(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)\cdot\boldsymbol{\tau}_0+\cdots \\ &\quad +\mathbf{P}^n(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)\cdot\boldsymbol{\tau}_0+\cdots \\ &= \frac{1}{1-\mathbf{P}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)}\cdot\boldsymbol{\tau}_0. \end{aligned} \tag{7}$$

Note that in (7), $E(\tau_R|\mathbf{s}_\chi,\mathbf{c}_\chi)$ was expanded as a summation of recursive retransmission costs multiplied by the corresponding probabilities, with an assumption that all processing cost upon data loss is ignorable. $\mathbf{P}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)$ is the vector of batch error probabilities with each element computed according to (2), and $\mathbf{P}^n(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi)$ is defined as

$$\begin{aligned} \mathbf{P}^n(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi) &= \mathbf{P}^{(n-1)}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi) \\ &\quad \cdot\mathbf{P}(\mathbf{s}_\chi+\mathbf{c}_\chi,\mathbf{c}_\chi), \quad n>1. \end{aligned}$$

The optimal distribution of parity-check packets, $\mathbf{c}_{\chi\text{opt}}$, is then given by

$$\mathbf{c}_{\chi\text{opt}}=\underset{\mathbf{c}_\chi}{\text{argmin}}\,E(\tau|\mathbf{s}_\chi,\mathbf{c}_\chi). \tag{8}$$

Eqs. (7)–(8) with (5) provide the theoretical solution to the problem. Yet an operational solution should also take into account the computation complexity as exhaustive search will not be feasible (the choice of $\mathbf{c}_\chi$ in (8) is arbitrary and therefore the solution space is not a close space). To accommodate real-time applications, a generalized *steepest decent* algorithm is used in this work to find the optimal (or a possibly suboptimal) solution. This fast algorithm starts with $\mathbf{c}_\chi=\emptyset$, i.e., no

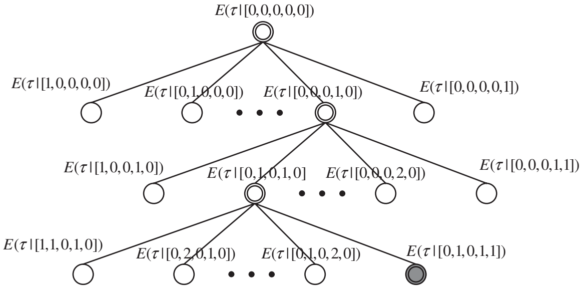

Fig. 5. Illustration on the steepest decent algorithm for finding $\mathbf{c}_{\chi\mathrm{opt}}$. The double-circled nodes indicate the searching path of the steepest decent algorithm, and the solid one represents the optimal operating point which has the minimum expected transmission delay.

parity-check packets are added at the initial point. At each iteration, the algorithm adds one more parity-check packet to either one of the χ batches, which results in χ possibilities. It then finds amongst the one that decreases the expected delay $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ the most, or stops the computation if $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ increases for all cases, where $E(\tau|\mathbf{s}_\chi, \mathbf{c}_\chi)$ is calculated using (7). Fig. 5 presents a simple illustration on this fast algorithm. The double-circled nodes indicate the searching path of the steepest decent algorithm, which have smaller expected transmission delays than their parents and siblings; the solid one represents the optimal operating point that has the minimum expected transmission delay, meaning that all its descendants (skipped in the plot) have larger expected delays. To simplify notation, the source vector $\mathbf{s}_\chi$ is ignored in Fig. 5 and only the vector of parity-check packets is indicated. From Fig. 5, it is apparent that the steepest decent algorithm has a linear computational complexity $O(K \cdot \chi)$, or more generally, $O(K \cdot L)$, where K is the total number of parity-check packets that are added (the depth of the tree) and L is the number of generated batches (the maximum width of the tree).

Fig. 6 presents a further illustration of the algorithm. For the Horse model (with 10 batches), it shows the computed minimum expected transmission delay with respect to the total number of parity-check packets that are added to the batches. In the computation, the packet size, the packet-loss rate, and the round-trip time are set to be $s = 500$ bytes, $p = 2\%$, and $r = 100$ ms, respectively. As can be seen in Fig. 6, adding parity-check packets at the beginning quickly decreases the expected transmission delay, which reaches a minimal at a certain point ($K = 9$ in this case). Thereafter, adding more parity-check packets gradually increases the expected transmission latency, implying that the additional transmission for sending parity-check packets has exceeded the potential retransmission cost.

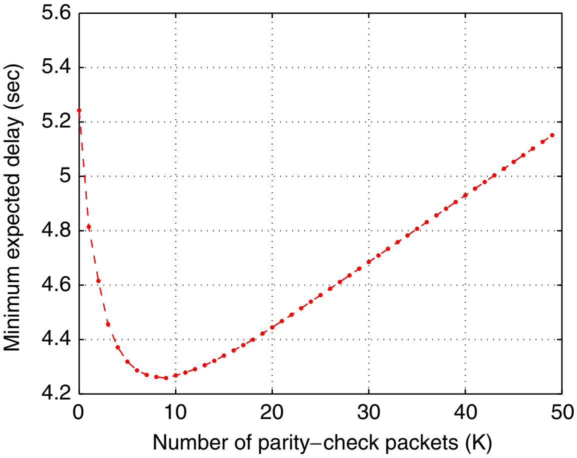

Fig. 6. The computed minimum expected delay versus the number of parity-check packets that are added. The computation is performed for the Horse model with a setting of these parameters: packet size $s = 500$ bytes, packet-loss rate $p = 2\%$, and round-trip time $r = 100$ ms.

To study the performance of the algorithm for different network conditions, we compute the optimal number of parity-check packets with respect to various packet-loss rates[4] and for different models. The results are shown in Fig. 7. The total number of parity-check packets that are needed for the minimum expected transmission delay is plotted with the packet-loss rate varying from 1% to 15%. As one may expect, when the network becomes heavily congested (higher packet-loss rates), the retransmission cost increases, and Fig. 7 shows that more parity-check packets are needed to minimize the expected transmission latency. An approximately linear relation is observed between the number of parity-check packets (K) and the packet-loss rate (p), which implies that the computation complexity is linearly increased as the packet-loss rate gradually increases, according to the expression $O(K \cdot L)$. In Fig. 7, it is noted that the Venus Head model requires more parity-check

[4] We keep the same round-trip time in all the computations based on the following fact: although the packet-loss rate increases quickly as the network becomes more congested, the round-trip time is observed with slight variation in all cases because of the regulation by RED gateways and the rate control mechanism.

packets (larger K) than the HORSE and DINOSAUR models with the same packet-loss rate, simply because its batches are encoded with larger sizes. In general, batch sizes can be reduced while a higher number of resolutions (larger L) will occur with potential loss/reduction in compression efficiency [23].

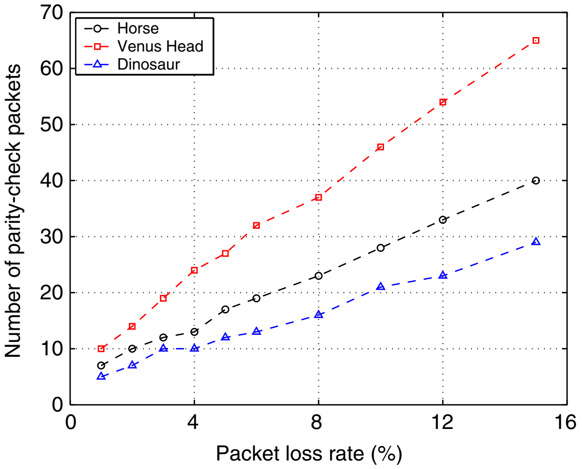

Fig. 7. The number of parity-check packets versus the packet-loss rate. The packet size and the round-trip time in the computation are set to be $s = 500$ bytes and $r = 100$ ms, respectively.

We now present simulation results of the proposed hybrid mechanism (REP) for distortion-constrained transmission, and compare it with the 3TP [5]. For fair comparison, in 3TP, we replace UDP with TFRC so the transport layer of 3TP is also TCP-friendly. Thus, in both REP and 3TP, the remaining data other than the selected batches are transmitted under the same mechanism, i.e., sole TFRC–UDP without FEC or retransmission. For this sake, in our presented results, we compare *only* the transmission delays for the batches that are selected to be transmitted reliably under a given distortion constraint, and are treated differently in 3TP and REP (in particular, 3TP uses TCP and REP uses hybrid UEP/SR).

Fig. 8(a)–(d) plots the transmission delay with respect to different numbers of batches that are selected to be reliably transmitted, determined upon the distortion constraint. The delay results have been averaged over all received meshes during the simulation period. The number of parallel flows in

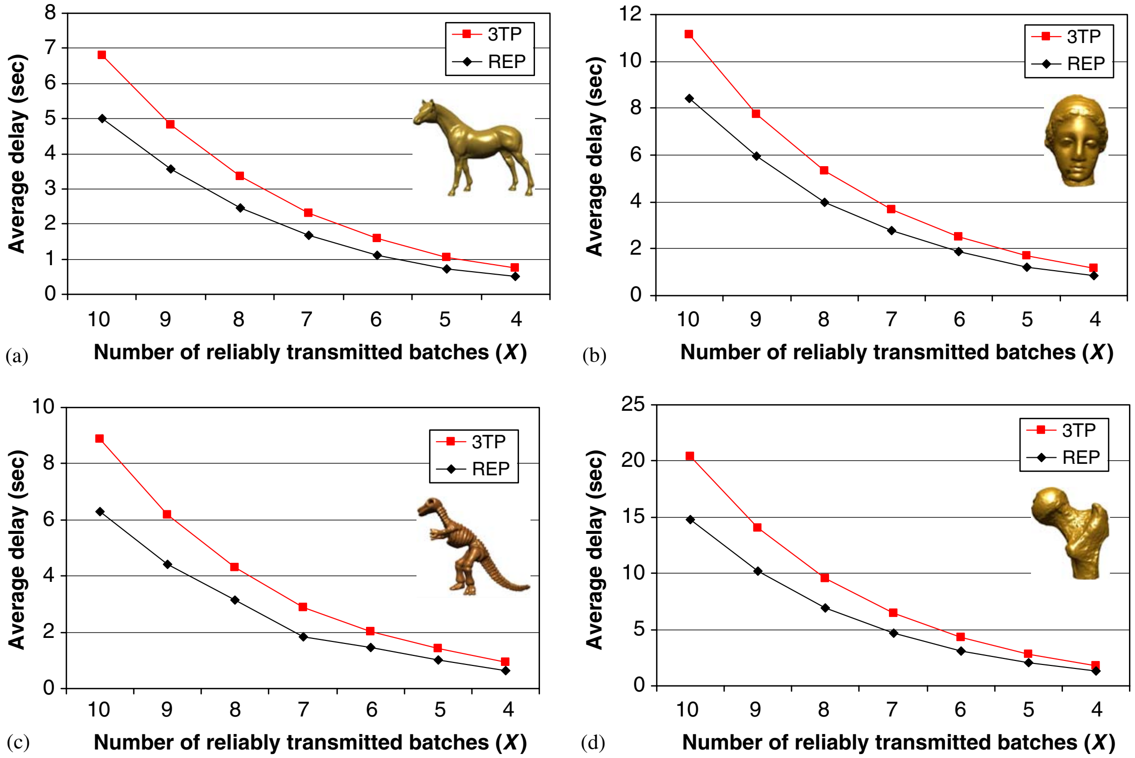

Fig. 8. Average transmission delays for sending various portions of data reliably using REP and 3TP, respectively. The number of parallel flows in the network is $f = 12$, and the packet sizes are $s = 1000$ bytes in (a) and (b) and $s = 500$ bytes in (c) and (d).

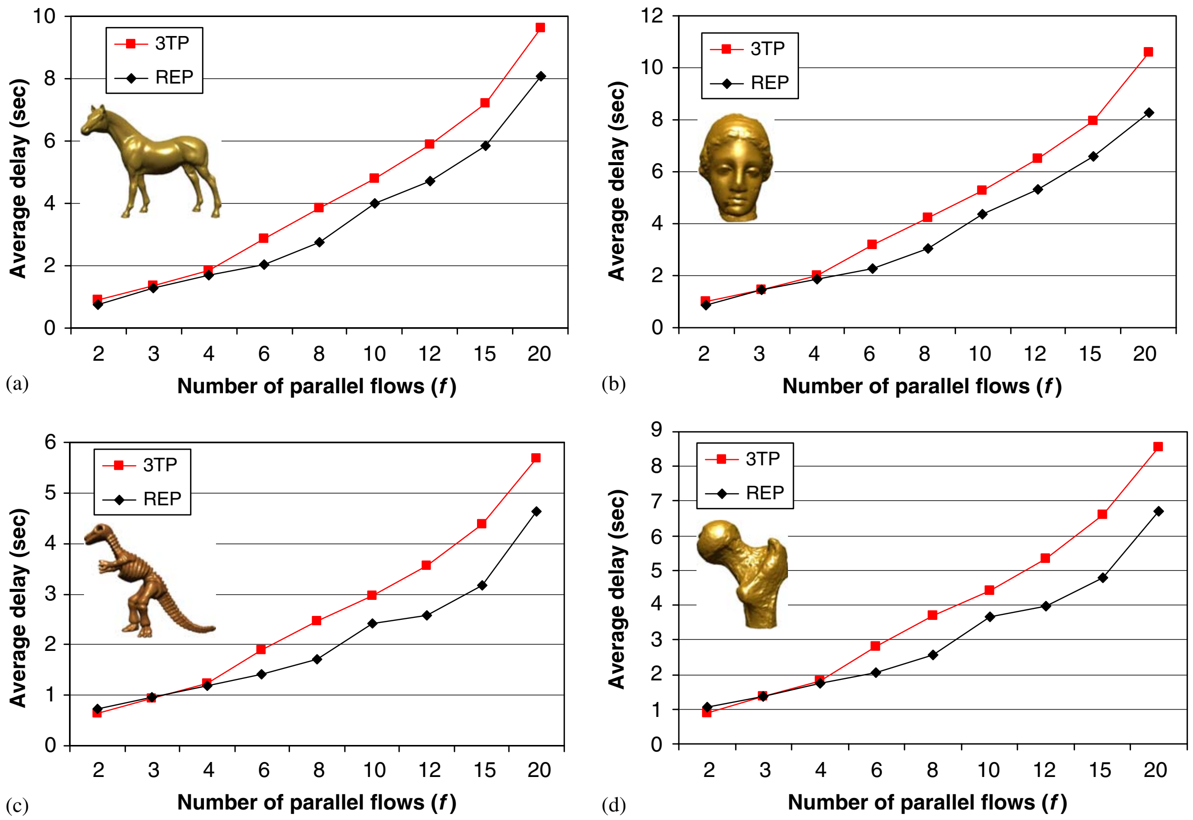

Fig. 9. Delay results when the distortion constraint $D_{\max}$ is 36 dB. The packet sizes are $s = 500$ bytes in (a) and (b) and $s = 1000$ bytes in (c) and (d). (a) $\chi = 8$, (b) $\chi = 7$, (c) $\chi = 9$, (d) $\chi = 8$.

the network is set to be $f = 12$, which results in a moderate network congestion situation as reflected by the measured packet-loss rate ($p \approx 6$–7%). The packet sizes are $s = 1000$ bytes for Fig. 8(a) and (b) and $s = 500$ bytes for Fig. 8(c) and (d), respectively. As can be seen from these plots, REP considerably reduces the transmission latency compared to 3TP. Specifically, when the full resolution model is required by the receiving application, all the enhancement batches need to be reliably transmitted ($\chi = 10$). Simply, 3TP returns to be sole TCP in this case. In contrast, REP still achieves substantial delay reduction by using hybrid UEP/SR. For example, in Fig. 8(a) and (b) where the packet size is $s = 1000$ bytes, 27% reduction in the average delay is observed for the Horse model and 25% is observed for the Venus Head model.

In Figs. 8(a)–(d), it is observed that as the portion of reliably transmitted data becomes smaller (i.e., smaller χ), which corresponds to a lower constraint on decoding distortion, the performance of REP gradually merges to 3TP. This behavior is anticipated because as the distortion constraint is lowered, both mechanisms transmit most of the data using TFRC–UDP without FEC or retransmission. Yet, considering a distributed online presentation of 3D scenes, a small reduction in the average delay provided by REP may still result in significant improvement on overall performance when a number of 3D meshes are transmitted on demand.

In Fig. 9, variation of the average delay for different network congestion situations is investigated. Suppose an upper bound of the decoding distortion, $D_{\max} \geqslant 36$ dB, is requested for the models to be rendered. To satisfy $D_{\max}$, $\chi = 7$–9 enhancement batches[5] are selected to be reliably transmitted according to Eq. (8) and Fig. 3. Fig. 9(a)–(d) presents the delay results of REP and 3TP for the selected batches for the test models. It is shown that REP outperforms 3TP under most of the network

[5] This number depends on the rate-distortion curve of the model.

conditions, and the gain is especially significant when the network encounters moderate or heavy congestion ($f \geqslant 6$ in the figures). For the network with light traffic load ($f \leqslant 4$), REP and 3TP perform similarly. When only two flows exist in the network and the packet size is $s = 1000$ bytes (Fig. 9(c) and (d)), one may note that 3TP provides slightly better results than REP. This is resulted from TCP's advantage of quick adaptation to the changes in available bandwidth compared with TFRC [13].

4.2. Delay-constrained transmission

So far, we detailed the proposed mechanism for distortion-constraint applications. In this part, we consider a time-critical situation where a delay constraint $\tau_{\max}$ is imposed by the application and a minimum decoding distortion is desired. A hybrid TCP/UDP protocol is not suitable for this problem because of the automatic retransmission behavior of TCP. Although having the application conservatively sends a small portion of data via TCP and the remainder via UDP may possibly satisfy the delay constraint, it is difficult to perform the *conservative* selection in TCP because it halves the transmission rate in response to a single packet drop, which results in frequent abrupt changes in throughput. In contrast, TFRC has lower variation of throughput over time [13], making it favorable in the delay-constraint application that is under consideration.

Utilizing the smooth sending rate variations in TFRC, we have developed in REP an operational algorithm with low computation complexity for delay-constrained transmission. In particular, for a given time frame, the algorithm selects the maximum number of batches to transmit provided that their minimum expected transmission latency does not exceed the allowable time frame. The minimum expected transmission latency is achieved by finding the optimal tradeoff between UEP and retransmission, and is computed using the proposed steepest decent algorithm as shown in the previous subsection. We consider the selected data (with the corresponding RS codes) as a *greedy conservative* solution at each transmission opportunity. After sending the selected source data and the corresponding parity-check packets, the server updates its sending buffer according to the feedback. The same procedure is repeated before all the batches are correctly received or the deadline has been reached.

We denote the source bit-rate of the L batches by vector $\mathbf{s}$, and use $\mathbf{s}^{(i)}$, $\tau^{(i)}$ to represent the source vector and the time instant for ith transmission opportunity, respectively. Major steps of the algorithm are then presented as follows:

(i) $i = 0$, $\mathbf{s}^{(0)} = \mathbf{s}$, $\tau^{(0)} = 0$;
(ii) $i = i + 1$;
(iii) update $\mathbf{s}^{(i-1)}$ to $\mathbf{s}^{(i)}$ by removing those batches that have been acknowledged;
(iv) if $\tau^{(i-1)} < \tau_{\max}$, find $(\chi, \mathbf{c}^{(i)}_{\chi\text{opt}})$ such that the following conditions are satisfied:

$$\begin{cases} 1 \leqslant \chi \leqslant L, \\ E(\tau^{(i)}|\mathbf{s}^{(i)}_{\chi}, \mathbf{c}^{(i)}_{\chi\text{opt}}) < \alpha \cdot (\tau_{\max} - \tau^{(i-1)}), \quad 0 < \alpha \leqslant 1, \\ \text{if } \chi < L, \text{ then } E(\tau^{(i)}|\mathbf{s}^{(i)}_{\chi+1}, \mathbf{c}^{(i)}_{\chi+1,\text{opt}}) > \alpha \cdot (\tau_{\max} - \tau^{(i-1)}), \end{cases} \tag{9}$$

where $\mathbf{c}^{(i)}_{\chi\text{opt}} = \operatorname{argmin}_{\mathbf{c}^{(i)}_{\chi}} E(\tau|\mathbf{s}^{(i)}_{\chi}, \mathbf{c}^{(i)}_{\chi})$ is given by Eqs. (7)–(8);
(v)send $(\mathbf{s}^{(i)}_{\chi} + \mathbf{c}^{(i)}_{\chi\text{opt}})$ using TFRC;
(vi) if $\tau_{\max}$ has not been reached, set $\tau^{(i)}$ with the current time; REDO (ii).

Note that in step (iii), $\chi \in [1, L]$ is the maximum number of batches with corresponding RS codes $(\mathbf{s}^{(i)}_{\chi}, \mathbf{c}^{(i)}_{\chi\text{opt}})$ whose expected transmission latency falls within the remaining time frame, $(\tau_{\max} - \tau^{(i-1)})$, multiplied by a fraction factor α. Ideally, we have $\alpha = 1.0$ given that smooth transmission throughput is provided by TFRC, in which case satisfaction of (9) can be considered as the greediest conservative selection of data that is to be sent. In practice, α may be chosen to be close (but not equal) to 1.0 for first few transmissions in order for the application to be more robust to abrupt drops of data sending rate occurred during transmission, thus avoiding large variation of the decoding distortion among received 3D models. The particular choice of the factor α is a design issue. In our simulation, we have found that a simple choice with $\alpha = 0.8$ for the initial transmission ($i = 1$) and $\alpha = 1.0$ for the rest transmissions ($i > 1$) works fairly well.

Performance of the above algorithm has been investigated compared with a heuristic mechanism that is also based on TFRC. It utilizes the time frame maximally by performing a simple scheme: the maximum number of batches that satisfy the timing condition are first selected based on the transmission throughput; the remaining time slot (if any) is then filled with the first few batches that have the least bit-rate but are generally most important. Pseudo-code of this simple filling algorithm (FA) is shown in the following, where T is the transmission throughput and $R(\chi)$ denotes the total bit-rate of χ batches, $\{B_1, B_2, \ldots, B_\chi\}$.

```
while $\tau < \tau_{\max}$ and $\mathbf{s}_\chi \neq \emptyset$ do
    find the maximum $\chi \in [1, L]$ s.t. $\frac{R(\chi)}{T} \leqslant \tau_{\max} - \tau$;
    send $\mathbf{s}_\chi = \{B_1, B_2, \ldots, B_\chi\}$ using TFRC;
    $\tau = \tau + \frac{R(\chi)}{T}$;
end while
```

Simulation results are presented in Figs. 10 and 11, where the number of parallel flows in the simulated network is set to be $f = 12$ (moderate traffic load) and the packet size is $s = 500$ bytes. Fig. 10(a)–(d) plot the decoding distortion averaged over all received meshes with respect to various delay constraints, i.e., different $\tau_{\max}$. One can see that REP greatly improves the decoding quality within the same time frame compared to the FA algorithm. In addition, as $\tau_{\max}$ increases, the decoding quality provided by REP quickly arises, whereas for FA, the resulting effect is observed to be unpredictable. It is a natural consequence of random losses as in the heuristic scheme, all the enhancement batches, except those few ones that are used to fill the remaining time slot, are treated equally without error resilience.

In Fig. 11, we trace the decoding distortion of every received Horse or Venus Head mesh for two individual cases: (a) and (b) $\tau_{\max} = 2$ s, and (c) and (d) $\tau_{\max} = 7$ s. The (blue) triangles marked curve denotes the PSNR$_m$ values of decoded meshes in REP, while the (red) circles marked corresponds to the FA mechanism. As expected, REP has much lower variation of the decoding quality in addition to a lower average distortion

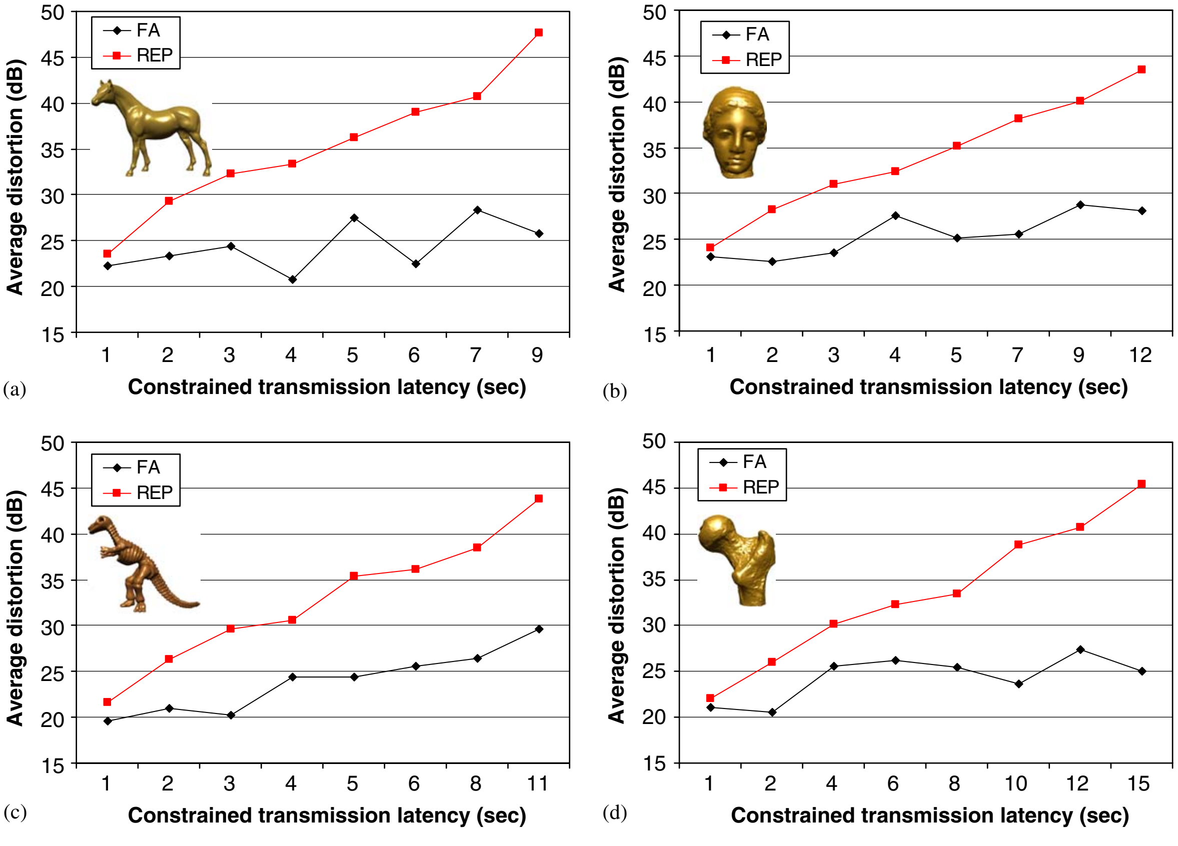

Fig. 10. Average decoding distortion of received meshes for various delay constraint in a typical network situation: the number of parallel flows is $f = 12$, the packet size is $s = 500$ bytes, and the resulting packet-loss rate is $p = 5.5\%$. (a) Horse; (b) Venus Hhead; (c) Dinosaur; (d) Balljoint.

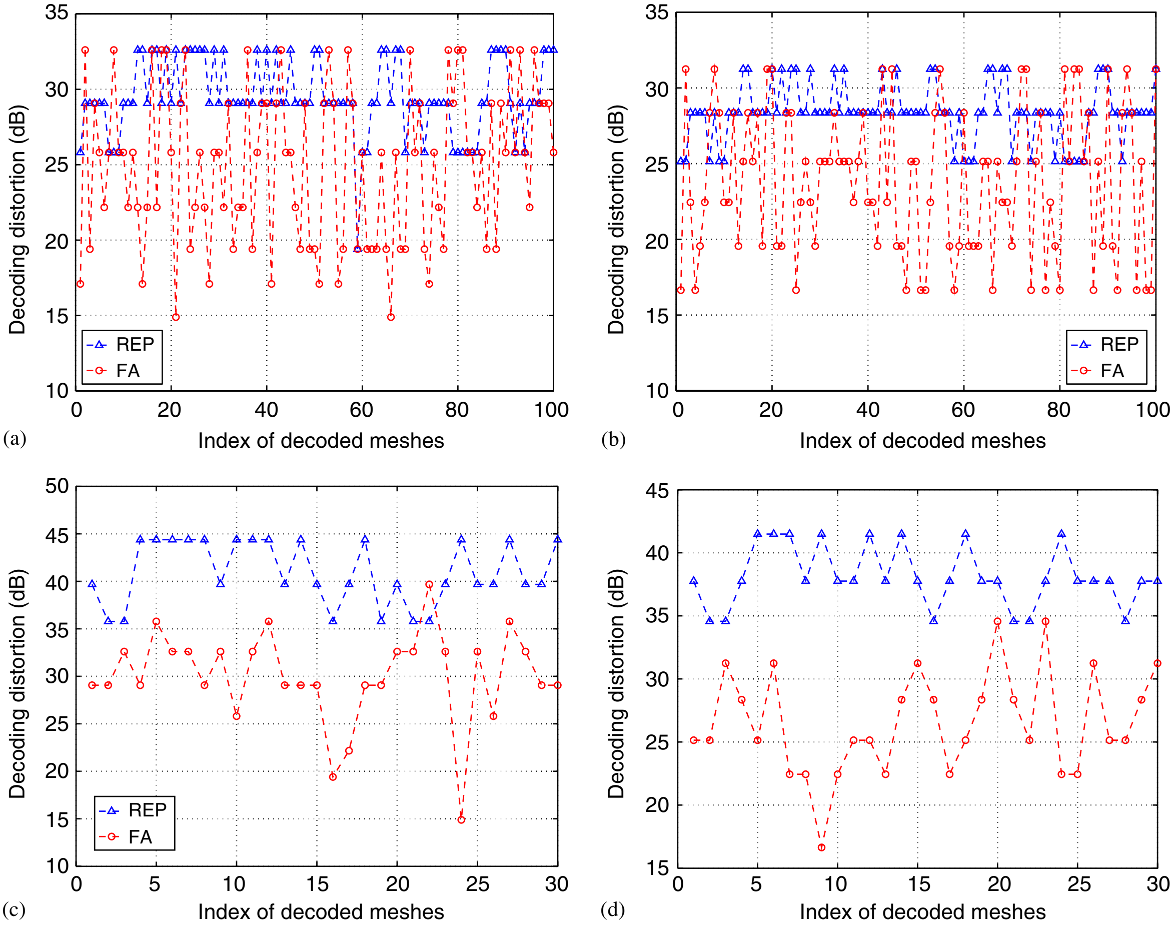

Fig. 11. Trace of the decoding distortion for each received mesh: (a) and (b) $\tau_{\max} = 2$ s; (c) and (d) $\tau_{\max} = 7$ s. (a) Horse, $\tau_{\max} = 2$ s; (b) Venus Head, $\tau_{\max} = 2$ s; (c) Horse, $\tau_{\max} = 7$ s; (d) Venus Head, $\tau_{\max} = 7$ s.

level (higher PSNR$_m$) than FA. Difference in the average quality is especially significant when larger transmission latency is allowed. For example, in Fig. 11(c) and (d), REP provides a median-level quality of 39.68 dB (for Horse) or 37.76 dB (for Venus Head), whereas FA has only 29.07 or 28.35 dB, correspondingly. Such difference in the average quality has actually been shown in Fig. 10. For the test models, average quality near to 40 dB is typically observed for REP when $\tau_{\max} = 7$–10 s, with a 10 dB or higher gain over the comparing heuristic. The perceptual quality difference of the rendered models is as significant as reflected by the quantities, captured by Fig. 12(a)–(d). All the above results, along with those presented in Section 4.1, have verified REP a successful transmission mechanism for streaming 3D meshes over a lossy network.

5. Conclusions and future work

We have proposed a hybrid unequal-error-protection and selective retransmission mechanism (REP) for streaming multi-resolution meshes over lossy networks with a goal of reducing response time and improving rendering quality in interaction for distributed graphics applications. In particular, to minimize the transmission latency under a distortion constraint, a portion of multi-resolution mesh data is selected to be reliably transmitted. Then, the best tradeoff between forward packet-loss resilience and retransmission is calculated in linear time using a steepest decent algorithm. To maximize rendering quality within a limited time frame, we developed an operational algorithm which has its core based on the proposed steepest decent algorithm. These algorithms have low computational

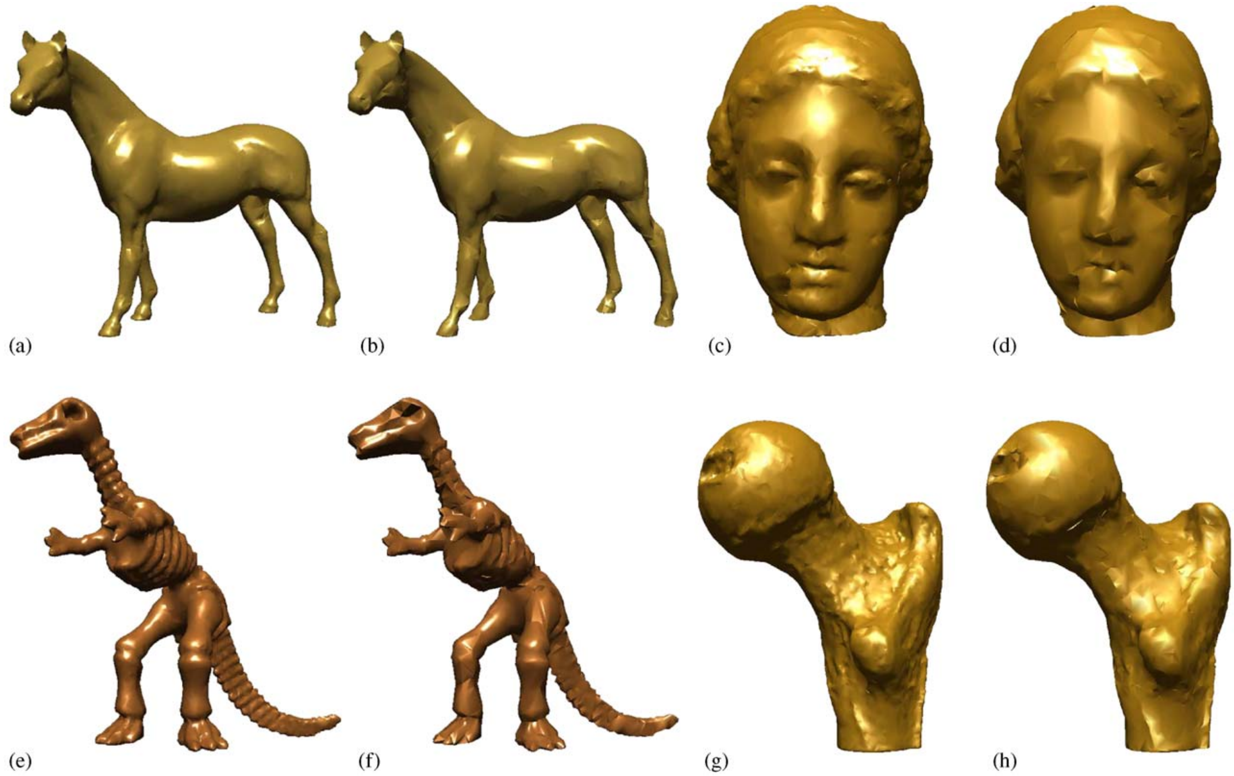

Fig. 12. Rendering quality near to 40 dB is obtained by REP when $\tau_{\max} = 7$–10 s, with a 10 dB or higher gain compared with the simple heuristic (FA) under the same situation.

complexity, making them attractive for distributed graphics applications.

TCP-friendly rate control (TFRC) is integrated in REP, which regulates the data sending rate in response to network congestion so that REP fairly competes for bandwidth with co-existing applications in the network. In addition, because TFRC has much lower variation of throughput over time than TCP, it is a favorable mechanism for REP to achieve smooth performance in interaction in time-critical situations.

REP is a network-oriented approach. It appropriately selects the transport and application layer techniques including TCP-friendly rate control, forward error protection, and feedback-based retransmission, and finds an optimal combination of them according to the characteristics of compressed meshes and the lossy channel. Pre- or post-processing methods such as data partitioning or error concealment are not included at this stage. While these techniques can be incorporated relatively independently (for example, the data partitioning technique provides error resilience by dividing a 3D scene/model into components, each of which can be encoded, transmitted, and decoded separately following the same process presented in the paper), exploration on their integration with the network-oriented approach is expected to further improve service quality for distributed graphics applications.

As yet, REP assumes that packet losses are primarily caused by buffer overflow (congestion) in wired networks. For streaming over wireless links where packets can be corrupted by channel errors at the physical layer, packet losses caused by bit errors should be considered. Not only rate control needs to be elaborated accordingly, but also an accurate model of the packet-loss process becomes important. Furthermore, the feedback channel is also error-prone because of the fading effect, which complicates the process of finding optimal transmission decisions. All these features are to be addressed to improve the performance for streaming 3D data over wireless channels.

Appendix A. Measuring distortion for multi-resolution meshes

The error between a simplified mesh surface S_1 and its original S_0 is generally defined as the

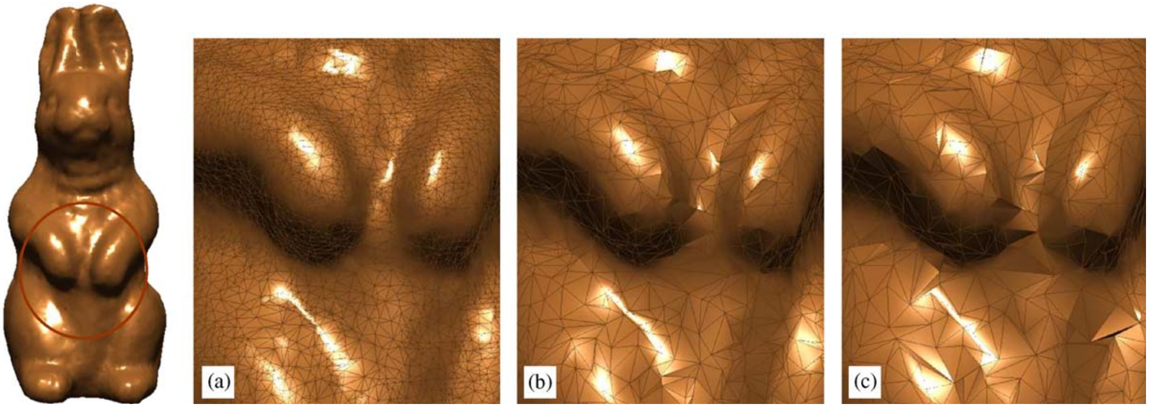

Fig. 13. An example when the Hausdorff (maximum) distance fails to reflect the quality degradation of different resolutions: (a) full resolution; (b) $\mathscr{E}_{\text{mn}} = 0.28$, $\mathscr{E}_{\text{rms}} = 0.18$, $H = 7.69$; (c) $\mathscr{E}_{\text{mn}} = 0.42$, $\mathscr{E}_{\text{rms}} = 0.36$, $H = 7.69$.

distance between corresponding sections of the meshes. Given a point v and a surface S, the distance $e(v, S)$ is defined as

$$e(v, S) = \min_{v' \in S} d(v, v'),$$

where $d()$ is the Euclidean distance between two points in $\mathbb{E}^3$. The one-sided distance between two surfaces S_0, S_1 is then defined as

$$\mathscr{E}_{\max}(S_1, S_0) = \max_{v \in S_1} e(v, S_0). \tag{10}$$

Note that this definition of distance is not symmetric, i.e., $\mathscr{E}_{\max}(S_0, S_1) \neq \mathscr{E}_{\max}(S_1, S_0)$ in general. The Hausdorff distance is a two-sided distance obtained by taking the maximum of $\mathscr{E}_{\max}(S_0, S_1)$ and $\mathscr{E}_{\max}(S_1, S_0)$, i.e., $H(S_1, S_0) = \max\{\mathscr{E}_{\max}(S_1, S_0), \mathscr{E}_{\max}(S_1, S_0)\}$. It has been observed though that the two one-sided distances usually have close numerical measures. In our definition, we preferably use (10) concerning that S_0 is the full-resolution mesh and S_1 is its simplified representation.

Similarly, the *mean* distance $\mathscr{E}_{\text{mn}}$ is defined as the surface integral of the distance divided by the area of S_1:

$$\mathscr{E}_{\text{mn}}(S_1, S_0) = \frac{1}{S_1} \int_{S_1} e(v, S_0)\, \mathrm{d}s. \tag{11}$$

And the *root-mean-square* summary $\mathscr{E}_{\text{rms}}$ squares each of the summed distances before normalization:

$$\mathscr{E}_{\text{rms}}(S_1, S_0) = \left(\frac{1}{S_1} \int_{S_1} e^2(v, S_0)\, \mathrm{d}s \right)^{1/2}. \tag{12}$$

All the above three error measures, i.e., (10)–(12), can be computed using the Metro tool [10], and are empirically verified in [26] successful predictors of visual quality as judged by human ratings and naming times.[6] In our experiments, we have confirmed that the mean and the mean-squared summaries properly measure the quality degradation of different resolutions for a multi-resolution mesh. However, it is noted that the maximum distance (or the Hausdorff distance), although commonly used in the literature, may not reflect the quality difference of consecutive levels properly. For example, a coarser resolution may have larger mean and mean-squared distance summary than its finer version whereas the Hausdorff distances for both levels are the same, as depicted in Fig. 13.

Furthermore, while the mean and mean-squared surface distances work successfully in measuring the quality of simplified meshes, they are not defined within meaningful or unified numerical ranges. The numerical values of the measured distances depend on particular coordinate grids that the mesh vertices lie on, and hence may drastically vary between meshes even in the same 3D scene or when the coordinate system is simply scaled. For example, the measured root-mean-square distance $\mathscr{E}_{\text{rms}} = 0.36$ in Fig. 13(c) does not convey any information about how much the mesh quality is degraded from its full resolution. To overcome this inconvenience for 3D applications which essentially deal with visualization, we define a PSNR-like metric for the multi-resolution mesh using the mean square distance summary. For a mesh with multiple resolutions $\{M_i\}_{i=0,\ldots,L}$, where M_0 is the base mesh and M_{L} is the full resolution, we

[6] Human ratings and naming times are two subjective measures widely used in the experimental sciences of visual fidelity.

define the quality of mesh M_i as

$$\mathrm{PSNR}_m \triangleq 20\log_{10}\frac{\mathscr{E}_{\max}(M_0, M_\mathrm{L})}{\mathscr{E}_{\mathrm{rms}}(M_i, M_\mathrm{L})} \quad (\mathrm{dB}), \qquad (13)$$

where $\mathscr{E}_{\max}$ and $\mathscr{E}_{\mathrm{rms}}$ are the measured maximum and mean-squared distances, respectively. Note that the distortion measure given by (13) no longer depends on the coordinate grids of the mesh. More

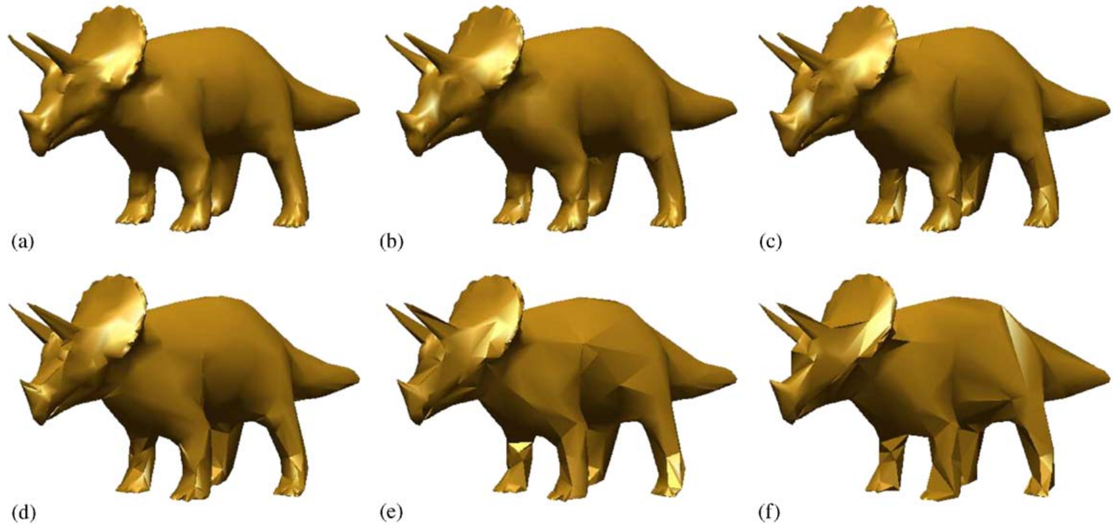

Fig. 14. Measured PSNR_m values for five resolutions of the Triceratops model.

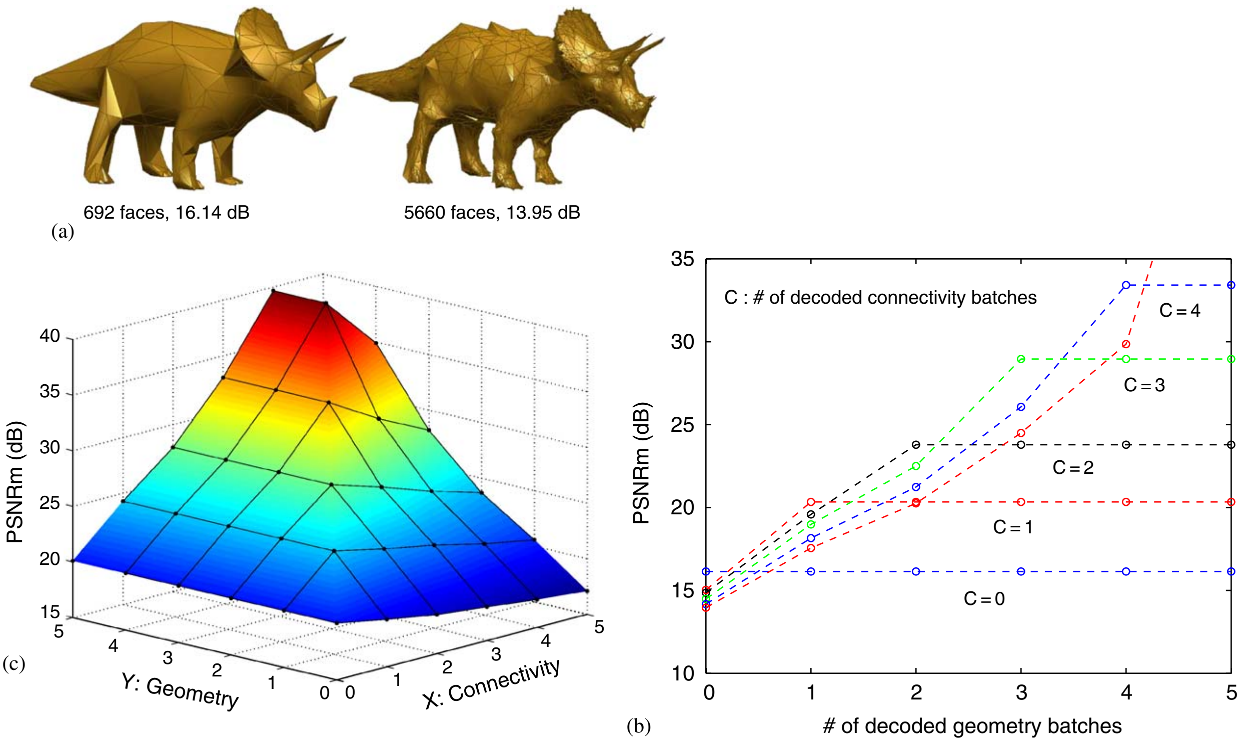

Fig. 15. For the Triceratops model: (a) decoding the mesh with sole connectivity data amplifies the geometric error and degrades visual quality; (b) and (c) the decoding distortion for different decoding patterns is measured by the PSNR_m, which objectively demonstrates the different impacts of connectivity and geometry on the decoder as well as the error-propagation effect of decoding connectivity without geometry. For example, the two models shown in (a) correspond to the highest and lowest points on the Y-axis in (b), respectively.

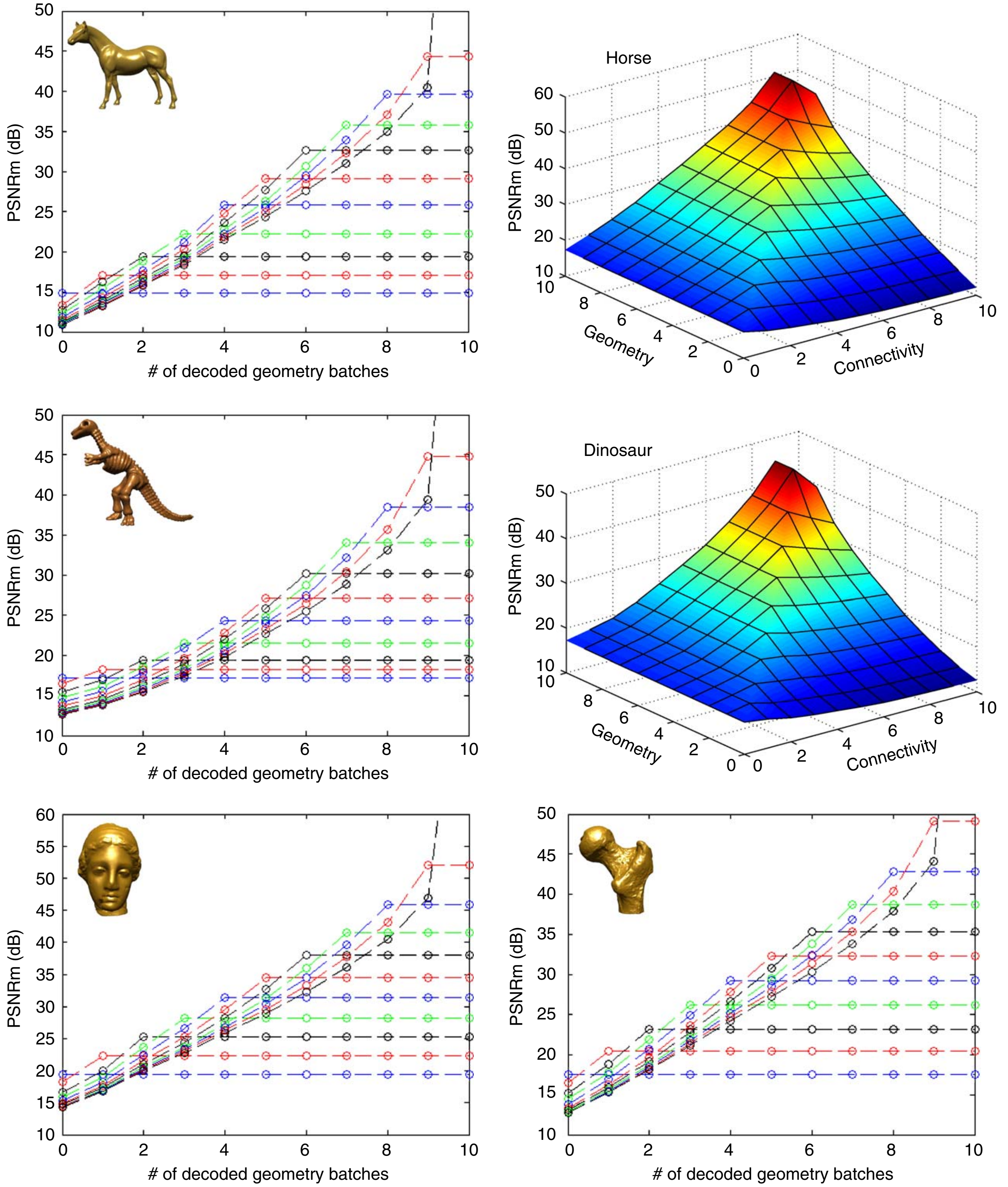

Fig. 16. Decoding distortion measured by PSNR$_m$ for different decoding patterns of the multi-resolution mesh. All the models are with 10 enhancement batches.

importantly, as shown in the paper, PSNR_m provides meaningful and approximately unified numerical values that well match human's subjective experience of using PSNR in imaging. Fig. 14 presents another example, where five resolutions are generated from the Triceratops model. One can see that the measured PSNR_m values meaningfully reflect the perceived difference of the resolutions compared to the full-resolution mesh.

Appendix B. Connectivity versus geometry

Connectivity and geometry information have different impacts on decoding multi-resolution meshes. Missing connectivity information stops the decoding process because the vertex-split process is not able to continue without knowing cut-edges; missing geometric data does not prevent the decoder from proceeding to the next level since the decoder can keep refining the topology using connectivity information with predicted vertex positions. For better understanding on this procedure, one may refer to [16,21] for further details.

An intuition resulting from the above description is that connectivity is more important than geometry. Nonetheless, it is observed in our experiments that, because of error propagation, decoding connectivity without refining geometry tends to amplify the geometric error and results in quality degradation instead of higher resolution. From a visualization point of view, it implies that both connectivity and geometry are crucial in preserving the quality. This conclusion has been utilized in the paper for designing the transmission mechanism. As a support, several results from our tests are presented in this section.

Fig. 15 gives an example of decoding the five resolutions of the Triceratops model. In Fig. 15(a), the base mesh with 692 faces and the decoded level with all faces added without geometric data are presented. One can see that the perceptual distortion of the latter model is substantially large compared with the base mesh even though it has the fully recovered topology. Note that the visual difference is also properly reflected by the PSNR_m values.

Fig. 15(b) presents more results for different decoding patterns using the PSNR_m metric. Each dashed line in Fig. 15(b) denotes a decoding path with a certain number of connectivity batches, noted as $C = 0, 1, 2, \ldots$ in the plot. For example, the line marked with $C = 0$ in Fig. 15(b) is horizontal because there is no connectivity information decoded, in which case all the geometry batches are also not decodable. On the other hand, the points that align vertically in the plot correspond to a decoding path with the same number of geometry batches and various connectivity. The base mesh in Fig. 15(a), for example, corresponds to the highest point on the Y-axis in the plot, while the lowest point on the Y-axis corresponds to the mesh with full connectivity as also shown in Fig. 15(a). Fig. 15(c) provides a three-dimensional perspective for the same results. The points on the diagonal of the mesh plot correspond to those on the envelope in Fig. 15(b), which have equal numbers of connectivity and geometry levels decoded. Conformably, such a point has either higher quality than the points that are on the same line with it in the X-dimension (same Y, larger X), or smaller data set compared with the points that horizontally connect to them in the Y-dimension (same X, larger Y). Similar results for the test models used in the paper are also obtained, which are presented in Fig. 16. All the results confirm that highest PSNR_m values are achieved by equally decoding connectivity and geometry.

References

[1] P. Alliez, M. Desbrun, Progressive compression for lossless transmission of triangle meshes, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, 2001, pp. 195–202.

[2] G. Al-Regib, Y. Altunbasak, An unequal error protection method for packet loss resilient 3-D mesh transmission, in: Proceedings of IEEE INFOCOM, vol. 2, 2002, pp. 743–752.

[3] G. Al-Regib, Y. Altunbasak, 3TP: an application-layer protocol for streaming 3-D graphics, in: Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 1, 2003, pp. 421–424.

[4] G. Al-Regib, Y. Altunbasak, J. Rossignac, An unequal error protection method for progressively compressed 3D models, IEEE Trans. Multimedia 7 (4) (2005) 766–776.

[5] G. Al-Regib, Y. Altunbasak, 3TP: an application-layer protocol for streaming 3-D graphics, IEEE Trans. Multimedia 7 (6) (2005) 1149–1156.

[6] G. Al-Regib, Y. Altunbasak, R.M. Mersereau, Bit allocation for joint source and channel coding of progressively compressed 3-D models, IEEE Trans. Circuits Syst. Video Tech. 15 (2) (February 2005) 256–268.

[7] C.L. Bajaj, S. Cutchin, V. Pascucci, G. Zhuang, Error resilient transmission of compressed VRML, Technical Report, TICAM, The University of Texas at Austin, 1998.

[8] B.-Y. Chen, X. Nishita, Multiresolution streaming mesh with shape preserving and QoS-like controlling, in: Proceedings of Web3D, 2002, pp. 35–42.

[9] Z. Chen, B. Bodenheimer, F.J. Barnes, Robust transmission of 3D geometry over lossy networks, in: Proceedings of Web3D, 2003, pp. 161–172.

[10] P. Cignoni, C. Rocchini, R. Scopigno, Metro: measuring error on simplified surfaces, in: Proceedings of Eurographics, vol. 17(2), 1998, pp. 167–174.

[11] S. Floyd, M. Handley, J. Padhye, J. Widmer, Equation-based congestion control for unicast applications, in: Proceedings of ACM SIGCOMM, 2000, pp. 43–56.

[12] S. Floyd, V. Jacobson, Random early detection gateways for congestion avoidance, IEEE/ACM Trans. Networking 1 (4) (1993) 397–413.

[13] M. Handley, S. Floyd, J. Padhye, J. Widmer, TCP friendly rate control (TFRC): protocol specification, Request for Comments (RFC) 3448, The Internet Society, January 2003.

[14] A.F. Harris, R. Kravets, The design of a transport protocol for on-demand graphical rendering, in: Proceedings of NOSSDAV, 2002, pp. 43–49.

[15] U. Horn, K. Stuhlmuller, M. Link, B. Girod, Robust internet video transmission based on scalable coding and unequal error protection, Signal Process: Image Commun. 15 (1999) 77–94.

[16] H. Hoppe, Progressive mesh, in: Proceedings of ACM SIGGRAPH, 1996, pp. 99–108.

[17] ISO/IEC JTC 1/SC 29/WG 11, Generic coding of audio-visual objects—Part 2: Visual, ISO/IEC 14496-2, 1999.

[18] A. Khodakovsky, I. Guskov, Normal mesh compression, in: Geometric Modeling for Scientific Visualization, Springer, Berlin, 2002.

[19] A. Khodakovsky, P. Schroder, W. Sweldens, Progressive geometry compression, in: Proceedings of ACM SIGGRAPH, 2000, pp. 271–278.

[20] J. Li, C.-C. Kuo, Progressive coding of 3D graphic models, in: Proceedings of the IEEE, 1998, pp. 1052–1063.

[21] R. Pajarola, J. Rossignac, Compressed progressive meshes, IEEE Trans. Visualization Comput. Graphics 6 (1) (2000) 79–93.

[22] R. Rejaie, M. Handley, D. Estrin, RAP: an end-to-end rate-based congestion control mechanism for realtime streams in the internet, in: Proceedings of IEEE INFOCOM, vol. 3, 1999, pp. 1337–1345.

[23] G. Taubin, 3D geometry compression and progressive transmission, in: Proceedings of Eurographics, 1999.

[24] G. Taubin, A. Gueziec, W. Horn, F. Lazarus, Progressive forest split compression, in: Proceedings of ACM SIGGRAPH, 1998, pp. 123–132.

[25] UCB/LBNL/VINT, network simulator ns (version 2), Available on URL: ⟨http://www.isi.edu/nsnam/ns/⟩.

[26] B. Watson, A. Friedman, A. McGaffey, Measuring and predicting visual fidelity, in: Proceedings of ACM SIGGRAPH, 2001, pp. 213–220.

[27] Z. Yan, S. Kumar, C.-C. Kuo, Error-resilient coding of 3-D graphic models via adaptive mesh segmentation, IEEE Trans. Circuits Syst. Video Tech. 11 (7) (2001) 860–873.