Joint Source and Channel Coding for 3-D Scene Databases Using Vector Quantization and Embedded Parity Objects

Dihong Tian, *Member, IEEE*, Junlin Li, *Student Member, IEEE*, and Ghassan AlRegib, *Member, IEEE*

***Abstract*—Three-dimensional graphic scenes contain various mesh objects in one geometric space where different objects have potentially unequal importance regarding display. This paper proposes an object-oriented system for efficiently coding and streaming 3-D scene databases in lossy and rate-constrained environments. Vector quantization (VQ) is exploited to code 3-D scene databases into multiresolution hierarchies. For the best distortion-rate performance, adaptive quantization precisions are allocated to different objects and different layers of each object based on a weighted distortion model. Upon transmission, scalably coded objects are delivered in respective packet sequences to preserve their manipulation independency. For packet loss resilience, a plurality of FEC codes are generated as "parity objects" parallel to graphic objects, which protect the graphic objects concurrently and also preferentially in regard to their unequal decoding importance. A rate-distortion optimization framework is then developed, which performs rate allocation between graphic objects and parity objects and generates the parity data properly. We show that, by treating graphic objects jointly and preferentially in source and channel coding while preserving their independencies in transport, the proposed system reduces the receiving distortion of the 3-D database significantly compared to conventional methods.**

***Index Terms*—Forward error protection, joint source and channel coding, multimedia, multiresolution, rate allocation, vector quantization.**

I. Introduction

Triangular meshes are the most widely used data format in computer generated graphics. A triangular mesh is represented by 3-D spatial vertices (called *geometry*) interconnected with arbitrary degrees (called *connectivity*). Similar to the image content, 3-D mesh objects have large volumes of data. Unlike images, which are naturally formatted as pixel grids, 3-D meshes require sophisticated rendering operations to be converted into pixel-based representations. Consequently, a 3-D scene database comprising pluralities of mesh objects requires considerable network bandwidth to be transmitted and computing power to be displayed on a remote terminal.

To alleviate the limitation, mesh compression has been investigated in the literature, and many algorithms have been proposed. Single-resolution compression such as [1] and [2] substantially reduces the number of bits needed to represent a 3-D model. To support scalable transmission and rendering, multiresolution coding algorithms, e.g., [3]–[7], were developed to code a 3-D mesh into a hierarchy of levels-of-details (LODs). For arbitrarily sampled meshes, multiresolution compression is generally performed by downsampling the mesh connectivity using *edge-collapse* operations [3], predicting the coordinates of collapsed vertices, and coding the prediction residuals along with the connectivity information that tells which edges should be split to recover the collapsed vertex.[1] Entropy coding was commonly used in coding coordinate residuals in separate spatial dimensions [3]–[6]. Vector quantization (VQ) was first introduced in single-resolution mesh compression [8], [9] to code the vertex geometry jointly. Recently, we studied the incorporation of vector quantization with a multiresolution hierarchy and proposed a VQM algorithm [10] for multiresolution mesh compression, which showed improved compression efficiency compared to its preceding algorithms that use scalar quantization and entropy coding.

All the existing algorithms focus on coding individual mesh objects. A 3-D scene, as aforementioned, synthesizes various objects in one geometric space. Depending on such factors as the objects' geometric complexities, interactions, and/or the application's semantics, different objects have potentially unequal contributions to the display quality. When coding the scene database under limited bit rates, such unequal display importance may be considered to present improved scene quality. To do so, one may apply the existing algorithms to code all objects separately into multiple resolutions and then select proper LODs for different objects such that the rate constraint is satisfied. Although this is a straightforward solution, it assumes a constant quantization precision for all the objects and for the entire LOD hierarchy of each object. Intuitively, with the same overall bit rate, higher scene quality may be achieved by coding more important objects or more important LODs of an object with higher quantization precisions and less important portions

Manuscript received May 2, 2006; revised February 18, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hassan Foroosh.

D. Tian is with Cisco Systems, Inc., San Jose, CA 95134 USA (e-mail: dihong.tian@ieee.org).

J. Li and G. AlRegib are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta GA 30332-0250 USA (e-mail: lijunlin@ece.gatech.edu; gregib@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2007.896696

[1]Wavelet-based algorithms (e.g., [7]) reported higher compression efficiency than the above methods by converting irregular meshes to be semi-regular before compression. In this paper, we preserve the input of irregular meshes and stick to the edge-collapse operation for LOD-hierarchy construction for two reasons: 1) irregularly sampled mesh is a more general format of 3-D geometric data, and 2) it does not require sophisticated preprocessing on the scene database.

with lower quantization precisions. In general, an adaptive coding scheme should intelligently determine the quantization precisions for different objects and different LODs of each object regarding their unequal decoding importance.

Adaptive quantization aims to properly distribute source bits among various LOD hierarchies to obtain improved distortion-rate performance. When a packet-lossy transmission environment is involved, two aspects regarding transmission efficiency and error resilience also need to be addressed. In one aspect, multiple objects are coded independently and, therefore, are desired to be delivered in respective packet sequences. Thus, at the receiving end, losing one packet will only corrupt or delay the decoding of a particular object, while other objects can still be decoded and manipulated. In the other aspect, unequal error resilience is desired for multiresolution objects to provide preferential error protection for more important objects as well as for more important layers of each object. The idea of unequal error protection (UEP) for hierarchical data was originated in [11]. Since then, UEP schemes were conventionally devised for single mesh or image object, e.g., [12]–[16], or multiple image objects in a separate fashion [17], [18]. Joint while preferential error protection for various multiresolution graphic objects has not been well addressed in the literature.

In this paper, we propose a joint source and channel coding system for 3-D scenes that properly accounts for all the above aspects. Compared to the preceding research on joint source and channel coding for scalable video transmission, e.g., [12] and [19], the presented work makes three contributions by addressing the unique properties of synthesized 3-D graphics. First, we propose an adaptive vector quantization scheme for coding a 3-D scene database into multiresolution hierarchies. Second, we propose an object-oriented streaming mechanism, which treats graphic objects jointly in error protection while preserving their independencies in transport. In doing so, we generate parity data as additional objects embedded in the stream of graphic objects. Finally, we develop a rate-distortion framework that allocates bit rates between source and parity objects and generates parity data toward optimal error resilience. Our experimental results demonstrate that the proposed system improves the quality of decoded 3-D databases significantly compared to conventional methods.

II. Overview and Preliminaries

A diagram of the proposed joint source and channel coding system is depicted in Fig. 1, which consists of two stages: an offline stage for multiresolution compression and an online stage for error-resilient streaming. We briefly describe these two stages in this section. Detailed studies on the major components will be presented in the following sections.

We consider an application where a plurality of 3-D objects are synthesized in one coordinate space. Regardless of the view perspective, the objects may have relative volumes depending on their coordinates, unequal geometric complexities, and different application semantics. These aspects are modeled by an object weighting module defined by the application. With such an application model, we exploit vector quantization to perform multiresolution scene compression. In doing so, we construct LOD hierarchies for the objects following progressive mesh

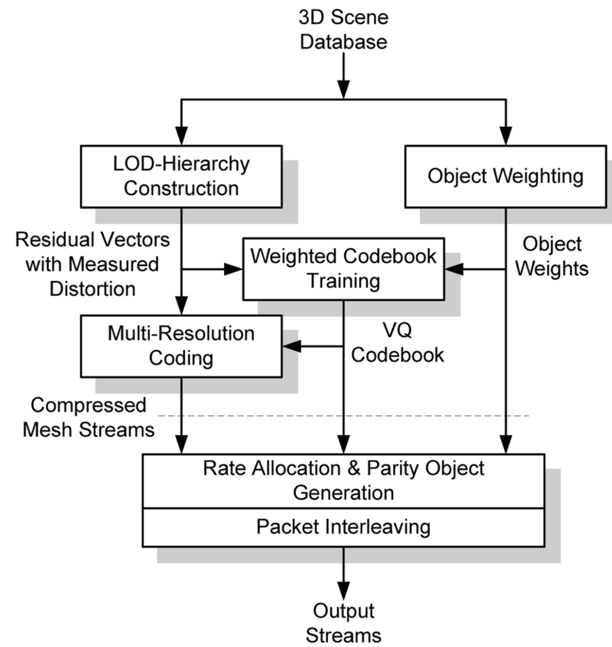


Fig. 1. Diagram of the presented joint source and channel coding system.

simplification [6] and encode the base meshes using single-resolution mesh compression methods such as those in [1] and [2]. To code the enhancement batches, we generate the VQ codebook using all the geometry prediction residuals produced by progressive mesh simplification as the training set. Moreover, each training vector is assigned a weight factor to reflect its relative importance in decoding the scene database. This weight factor is determined by incorporating application-dependent object importance with measured distortion in the mesh simplification process. A weighted codebook training algorithm is then executed, and the enhancement batches of the objects are compressed using thus produced codebook.

Upon transmission, the compressed base meshes are first transmitted through a reliable channel for a prompt display. Because the base layers in general have a fairly small fraction (1%–2% or less) of the entire bitstream, the network is mainly loaded by the transmission of the remaining data, which is the focus of our discussion in this paper. In particular, the compressed mesh streams are organized into respective packet sequences to preserve their manipulation independency during packet delivery. In contrast to the independent transport, for error resilience, the packets of graphic objects are protected concurrently while also preferentially by a plurality of FEC codes. The parity data of each FEC code is treated as a separate object parallel to graphic objects. Based on weighted distortion-rate properties, an optimization framework performs rate allocation between graphic objects and parity objects and generates parity data correspondingly. Finally, all the objects are transmitted in an interleaved manner to allow equally fast access to each object at the receiving end.

A. Object Weighting

As the first component of the presented coding system, object weighting assigns to each object a weight factor that is expected to reflect the relative importance of the object in the

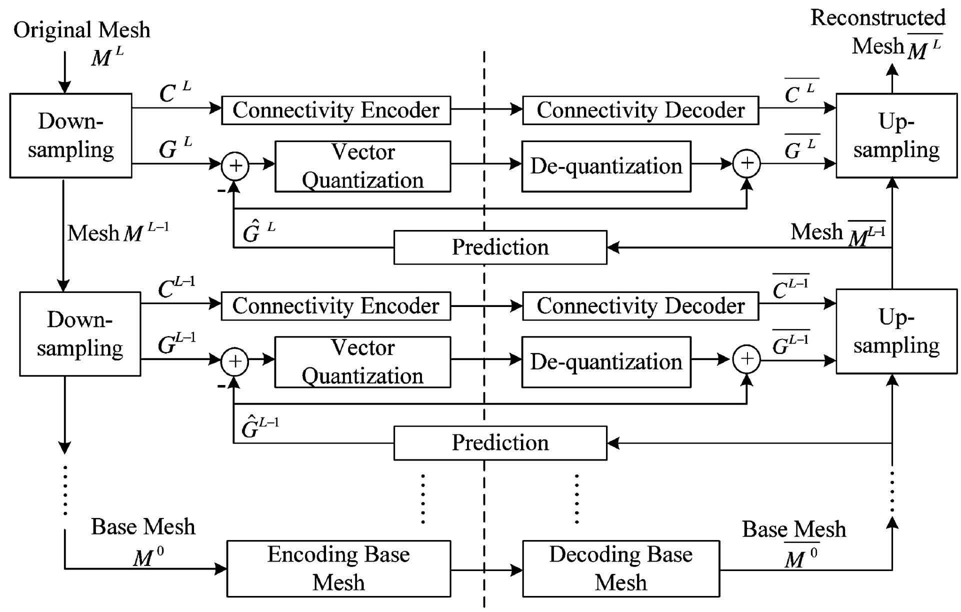


Fig. 2. Diagram of the LOD-hierarchy construction/reconstruction process and the VQ-based multiresolution coding for one mesh object.

decoded scene database. The determination of objects' weight factors will not only account for the relative volumes of the objects in the synthesized coordinate space and the visual (geometric) complexity of the objects, but also their inherent importance based upon the semantics of the application.[2] These features are essential to the coding of objects and are independent from the manipulating view perspective. Although a view-dependent treatment may be introduced in the streaming mechanism, it is not considered in the presented work. In general, modeling the relative importance of the objects is in general application specific. For this reason, we do not intend to specify an object weighting process in this paper. Rather, we assume that each object κ in a given scene database, $\mathcal{S}$, has been assigned a weight factor, $\omega_\kappa \in [0,1]$, to reflect its application-specific importance. All the weight factors are normalized so that their summation is equal to 1, i.e.,

$$\sum_{\kappa \in \mathcal{S}} \omega_\kappa = 1. \quad (1)$$

In the rest of the paper, we discuss the other components of the proposed coding system based on given object weights.

B. LOD Hierarchies

In parallel to the object weighting, a multiresolution hierarchy is constructed for each mesh object in the input scene database, following a process similar to [10]. To be explicit, Fig. 2 shows a diagram of the multiresolution coding procedure for a single object, which can be summarized in three steps. 1) The full-resolution mesh M^L is downsampled to generate a sequence of LODs, $M^{L-1}, \ldots, M^0$, by performing successive *half-edge* collapse operations [3] at each level. Every half-edge collapse operation merges one vertex of an edge to the other, which alters the neighborhood of the collapsed vertex. The corresponding connectivity information of the collapsed vertices, $\mathcal{C}^L, \mathcal{C}^{L-1}, \ldots, \mathcal{C}^1$, are encoded [6] following a traverse order, while the geometry data, $\mathcal{G}^L, \mathcal{G}^{L-1}, \ldots, \mathcal{G}^1$, are buffered. 2) The base mesh is encoded using single-resolution mesh compression algorithms, e.g., [2]. 3) Starting from the decoded base mesh, the enhancement geometry data, $\mathcal{G}^1, \mathcal{G}^2, \ldots, \mathcal{G}^L$, are sequentially predicted based on previous reconstructed LODs, and the prediction residuals are compressed by vector quantization. Different from the existing algorithm [10] which employs an independent training set to generate the VQ codebook, in this paper we exploit a scene-adaptive vector quantization scheme to compress the geometry residuals, which is detailed in Section III.

For a single object, the *normalized distortion* of an LOD M^i in the constructed LOD hierarchy is measured by

$$\mathcal{D}(M^i) = \frac{\mathcal{E}_{\mathrm{rms}}(M^i, M^L)}{\mathcal{E}_{\mathrm{max}}(M^0, M^L)} \quad (2)$$

where $\mathcal{E}_{\mathrm{rms}}(M^i, M^L)$ and $\mathcal{E}_{\mathrm{max}}(M^0, M^L)$ are the root-mean-square distance and the maximum distance, respectively, between the corresponding pairs of mesh surfaces. These error measures can be calculated using the fast Metro tool [20] in practice. The normalized distortion captures the quality difference between different layers within the LOD hierarchy of each object, with the assumption that the difference between objects has been modeled by the weight factors assigned by the application. The number of LODs generated for each object is determined based on its geometric complexity, which ensures that a similar maximum error level will be used for normalizing the distortion of LOD hierarchies.

Now, given a scene database with N multiresolution objects, $\{M_\kappa^i\}, \kappa = 1 \ldots N, i = 0 \ldots L_\kappa$, and the object weights, $\{\omega_\kappa\}$, we measure the distortion of a decoded scene by

$$\mathcal{D}_s\left(\{M_\kappa^{i_\kappa}\}\right) = \sum_{\kappa=1}^{N} \omega_\kappa \cdot \mathcal{D}\left(M_\kappa^{i_\kappa}\right), \quad 0 \le \omega_\kappa \le 1 \quad (3)$$

[2]For instance, merchandise should be more important than shelves in a virtual shop, and paints would be more important than walls in a virtual gallery.

where $\mathcal{D}(M_\kappa^{i_\kappa})$ denotes the measured distortion for mesh M_κ decoded at a resolution i_κ.

III. Vector Quantization

We exploit vector quantization to compress enhancement geometry data following the hierarchical predictive structure depicted in Fig. 2. In [10], vector quantization was adopted for the multiresolution compression of single mesh objects, where the VQ codebook is generated by using a separate set of training models and the codebook training algorithm is independent from the coding process. Such a codebook independency is necessary for coding a single mesh object as otherwise the codebook must be transmitted with the object, which can incur a considerable overhead to the bitstream. When coding a scene database comprising a number of mesh objects, however, the overhead of the codebook is small compared to the entire bitstream. Meanwhile, using prediction residual vectors generated from all mesh objects in the scene database as the training set of the codebook is expected to yield improved compression efficiency. Furthermore, by partitioning the training vectors into various sets in accordance with their decoding importance, adaptive quantization precisions can be allocated to different objects as well as different LODs of each object.

Based on the discussion above, we develop a weighted training algorithm to produce the VQ codebook for compressing the geometry data of a scene database. The algorithm is adaptive in three senses: 1) it is adaptive to the contents of the input 3-D scene database, as the training set is taken from all objects in database, 2) it is adaptive to the application-specified object importance, and 3) it is adaptive to the decoding importance of the different LODs of each object.

A. Weights of Training Vectors

The proposed codebook training process takes prediction residuals from all mesh objects in the input scene database as training vectors. We denote the entire set of training vectors by a union of subsets, $\bigcup_{\kappa,i_\kappa}\{\mathbf{x}_{\kappa,j}^{i_\kappa}\,|\,j=1,2,\ldots,n_\kappa^{i_\kappa}\}$, where each subset $\{\mathbf{x}_{\kappa,j}^{i_\kappa}\}$ is a batch of $n_\kappa^{i_\kappa}$ residual vectors for decoding mesh $M_\kappa^{i_\kappa}$ toward its higher LOD $M_\kappa^{i_\kappa+1}$. To account for the unequal importance of different objects and different LODs of each object in vector quantization, we assign each training vector $\mathbf{x}_{\kappa,j}^{i_\kappa}$ a weight factor, $\gamma(\mathbf{x}_{\kappa,j}^{i_\kappa})$, which incorporates both the object weight and the measured distortion of the corresponding LOD. In particular, we have

$$\gamma\left(\mathbf{x}_{\kappa,j}^{i_\kappa}\right)=\omega_\kappa\cdot\frac{\Delta D_\kappa^{i_\kappa}}{n_\kappa^{i_\kappa}} \tag{4}$$

where ω_κ is the weight of the object and

$$\Delta D_\kappa^{i_\kappa}=D\left(M_\kappa^{i_\kappa}\right)-D\left(M_\kappa^{i_\kappa+1}\right) \tag{5}$$

denotes the *reduction* of normalized distortion when the higher LOD $M_\kappa^{i_\kappa+1}$ is successfully decoded.

Two heuristics are implied by the weight allocation given in (4). First, vectors of lower LODs are allocated larger weights than those of higher LODs for the same object. This is because decoding lower LODs in general results in more significant distortion reduction (larger $\Delta D_\kappa^{i_\kappa}$) than decoding higher LODs.

(i) Code vector initialization:

$$y_1^{(1)},\cdots,y_{2^q}^{(1)},m=1.$$

(ii) Nearest neighbor repartition ($1\le i\le K$):

$$j=\arg\min_{1\le l\le 2^q}\left\|\mathbf{x}_i-\mathbf{y}_l^{(m)}\right\|, \quad \textit{Put } \mathbf{x}_i\to\mathcal{R}_j^{(m)}, \tag{6}$$

$$\mathcal{D}_m=\mathcal{D}_m+\gamma_i\cdot\left\|\mathbf{x}_i-\mathbf{y}_j^{(m)}\right\|^2. \tag{7}$$

(iii) Centroid computation ($1\le l\le 2^q$):

$$\mathbf{y}_l^{(m)}=\sum_{i:\mathbf{x}_i\in\mathcal{R}_l^{(m)}}\gamma_i\cdot\mathbf{x}_i\Big/\sum_{i:\mathbf{x}_i\in\mathcal{R}_l^{(m)}}\gamma_i. \tag{8}$$

(iv) Code vector jiggling ($1\le l\le 2^q$):

$$T_m=\sigma_x^2\left(1-\tfrac{m}{I}\right)^3, \quad \mathbf{y}_l^{(m)}=\mathbf{y}_l^{(m)}+\xi_l(T_m). \tag{9}$$

(v) Stopping condition:

$$\textit{If } m\ge I,\ \textit{Stop}, \quad \textit{Otherwise, } m=m+1,\ \textit{Goto (ii)}. \tag{10}$$

Fig. 3. Weighted codebook training algorithm.

Second, within each batch, all training vectors have the same weight. In other words, they will be treated equally importantly in minimizing quantization errors in the codebook training process.

B. Codebook Training Algorithm

By introducing vector weights we aim to quantize various batches of geometry data with different precisions to attain higher distortion-rate performance. To this end, we develop a weighted codebook training algorithm based on the stochastic relaxation (SR) algorithm proposed in [21]. In the following, we present a complete description of the weighted codebook training algorithm. In our presentation, we consider a q-bit vector quantizer. Therefore, the codebook contains 2^q code vectors. We use $\{\mathbf{x}_i,\gamma_i\}, i=1,2,\ldots,K$, to denote the entire set of training vectors and their corresponding weights allocated by (4).

The major steps of the codebook training algorithm are shown in Fig. 3, which are performed in an iterative fashion. In the algorithm, $\mathcal{R}_j^{(m)},\mathbf{y}_j^{(m)}$, and $\mathcal{D}_m$ denote the jth partition region, the jth code vector, and the calculated distortion, respectively, at the mth iteration. The procedure of code vector jiggling, as given in (9), is used to prevent the algorithm from staying in poor local optimal results [21]. In particular, $\xi_l(T_m)$ denotes a perturbation noise added to the code vector, and the noise variance is dictated by the so-called temperature T_m, which gradually decreases as m increases. σ_x^2 indicates the variance of the code vector components, and finally, I defines the number of iterations to be run. More detailed discussions on these parameters can be found in [21].

The presented codebook training algorithm differs from the SR algorithm [21] by accounting for vector weights in discrepancy evaluation (7) and centroid computation (8). As a result, the algorithm tends to enclose more important vectors in smaller partitions to reduce their quantization errors. Although this weighted training process is not theoretically justified to yield optimal results, its effectiveness is confirmed by empirical success, as will be shown in Section VI.

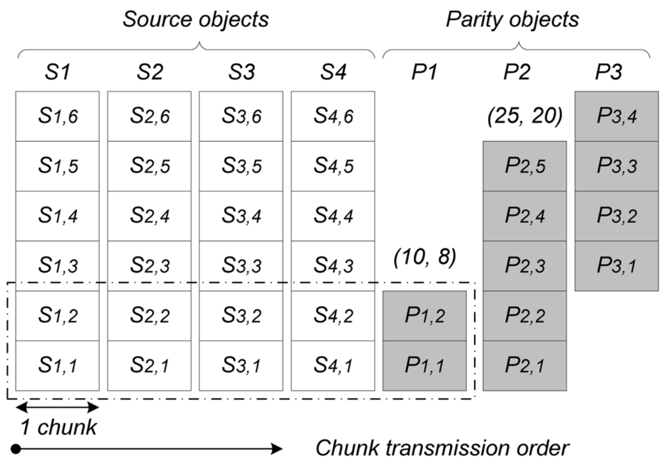


Fig. 4. Illustration on the chunk-based transmission and the use of parity objects for joint unequal error protection. Note that the transmission of multiple objects is interleaved to allow equally fast access to the objects.

IV. Embedded Parity Objects

One essence of the adaptive vector quantization is the preferential treatment on coding the LOD hierarchies of various mesh objects with respect to their unequal importance. When we consider error-resilient delivery of the scalably coded scene database under a lossy and rate-constrained environment, it is natural to embrace the same spirit and devise unequal error protection (UEP) schemes. Unlike conventional UEP schemes designed for single mesh or image objects [12]–[16], the change from a single-object context to a multiobject one invokes two fundamental aspects that need to be properly addressed together with UEP. In one aspect, because multiple mesh objects are coded independently, they are desired to be delivered in respective transport sequences to preserve the independence at the receiving application. In the other, with the same amount of parity redundancy, the highest error-protection efficiency is expected to be accomplished by joint considerations on multiple objects. We address these issues with UEP in this section.

A. Partially Ordered Streaming

Earlier transmission schemes proposed for multiobject media applications organize the bitstream of multiple objects into a strictly ordered packet sequence, where each packet contains various numbers of bytes for the objects [17], [18]. In those cases, packetization is essentially byte-oriented, and one lost packet may corrupt or delay (if retransmission is allowed) the decoding of multiple objects. To address the coding independency properly, a partially ordered packetization scheme is employed in the proposed transmission mechanism. As shown in Fig. 4, instead of packetizing the entire bitstream into a strict sequence, the bitstream of each object is respectively organized into a sequence of *chunks*. At the sending side, chunks of multiple objects are transmitted in an interleaved or round-robin fashion. At the receiving end, multiple chunks from different objects may be received in arbitrary orders, as far as the respective sequence of each object is maintained.

The respective order of each object ensures that the manipulation independency of multiple objects is well preserved in packet delivery under random network behaviors, as in this case, one lost or delayed packet will only affect the sequenced data reception and decoding of a particular object, leaving no impact on the reception or decoding of other objects.

B. Joint Unequal Error Protection

We now study chunk-level forward error correction (FEC) for the error resilience of multiple objects. To this end, we consider common FEC codes such as Reed-Solomon codes, where for every block of k data chunks $h=(n-k)$ parity chunks are generated to form an n-chunk FEC block. Such an (n,k) systematic code can recover up to h chunk losses (erasures), as positions of the lost chunks in the chunk sequence are known. Principles and implementation issues of such block erasure codes can be found in many literatures, e.g., [22].

One natural implementation of the chunk-level FEC is to append parity chunks to data chunks for different objects, respectively, similar to the schemes in [17] and [18]. Although such separate FEC can easily implement unequal error protection for multiple objects, as well as multiple layers of each object, it is not efficient as one parity chunk only provides error resilience to a certain object while it introduces redundancy to all the objects. In this paper, we propose to use "parity objects" to overcome the deficiency. Each *parity object* with a size of h chunks refers to an (n,k) systematic code where k data chunks from one or many objects are protected by $h=n-k$ parity chunks. Upon transmission, this parity object is assigned a separate object identification number (OID), as shown in Fig. 4. Since both data and parity chunks are uniquely indexed, the receiving end can recognize the locations of lost chunks and recover the code block under the code's correcting capability. Meanwhile, partially ordered transport are still preserved for all the streams. In addition, with parity objects, unequal error protection can be implemented for arbitrary sets of data chunks by generating multiple parity objects. In Fig. 4, for example, chunks $(S_{i,j},P_{1,j}), i=1,2,3,4, j=1,2$, form a $(10,8)$ FEC code, and $(S_{i,j},P_{2,j}), i,j=1,2,\ldots,5$ form a $(25,20)$ code. Same as source objects, parity objects are respectively ordered, and are embedded into the chunk stream through interleaving.

It is evident that, with the same amount of redundance, using parity objects can present concurrent and, hence, improved error resilience to multiple source objects compared to separate FEC. We use a scenario of equal error protection to clarify this statement, where for a number of s source objects, separate FEC appends h parity chunks for every block of k data chunks in each object, respectively, while a parity object generates hs parity chunks for the total number of ks data chunks concurrently. In both cases, the entire FEC block under consideration has a total number of ns chunks. Let $\gamma_I(s), \gamma_P(s)$ denote the probabilities that the entire block is correctly decoded in the separate-FEC and the parity-object cases, respectively. We have $\gamma_I(s) = \Pr\{\leq h \text{ losses out of } n \text{ chunks in each of } s \text{ objects}\}$, and $\gamma_P(s)=\Pr\{\leq hs \text{ losses out of } ns \text{ chunks}\}$.

Proposition 1: Independent from the random loss process in the transmission of the ns-chunk block, $\gamma_I(s) \leq \gamma_P(s)$ for $s \geq 1$.

Proof: Divide the ns-chunk block into s groups according to the transmission sequence, where chunks with index i (mod n) belong to the same group. Suppose that there are

$\nu_1, \nu_2, \ldots, \nu_s$ $(0 \leq \nu_i \leq n, i = 1, 2, \ldots, s)$ losses out of n chunks in each group. In the separate FEC case, the event that the ns-chunk block is recoverable is equivalent to $\nu_i \leq h$ for $i = 1, 2, \ldots, s$. Hence, $\sum_{i=1}^{s} \nu_i \leq hs$, which means that the entire block is also recoverable with an hs-chunk parity object regardless of the loss pattern in each group. The inverse of the above reasoning is not true. Therefore, $\gamma_I(s) \leq \gamma_P(s)$. ■

Proposition 1 presents a general comparison regarding the recovery probability of the entire data block. Nevertheless, without the proper generation of parity objects, there exist scenarios where individual objects may still be decodable with the separate FEC while parity objects fail to decode the block due to a high rate of chunk losses. To provide a solution to this problem, in the next section, we devise a parity-object generation algorithm, which generates parity objects properly based on source and channel statistics. Empirically, we show that by taking advantage of the flexibility provided by parity objects, the parity-object generation algorithm improves error resilience considerably compared with separate FEC.

It should be mentioned that using a larger FEC block will require increased coding complexity, which may be upper bounded by resource limitations in a real implementation. Beside that, in both FEC schemes, certain side information needs to be transmitted for the receiver to decode the FEC codes correctly. Efficient coding of such side information is not studied in this paper. Instead, we assume that overhead for sending the side information is negligible compared to the entire bitstream. In the rest of the paper, we use triple $(n, k, \mathbf{I})$ to represent an (n, k)-code parity object, with $\mathbf{I}$ denoting the indices of the k data chunks. Apparently, $k = |\mathbf{I}|$ is the size of set $\mathbf{I}$.

V. Joint Source and Channel Coding

While the larger FEC block, in general, provides the higher error-correcting ability, data chunks of graphic objects have unequal decoding importance and, hence, desire preferential error resilience to achieve optimized rate-distortion performance. Next, we study in detail how parity objects should be generated for a set of scalably coded graphic objects and how a given rate budget should be distributed amongst all the source and parity objects, with the goal of providing maximized expected receiving quality. More explicitly, we aim to tackle the following problem.

Given a set of N scalably coded mesh objects with preassigned object weights $\{\omega_\kappa\}_{\kappa=1}^{N}$, an index set $\mathbf{I}_N$ for all data chunks, a chunk-erasure channel, and a rate constraint $\mathcal{R}_\mathcal{C}$ measured in chunks, find a rate allocation $\Psi = \{s, c; \mathbf{I}_s\}, \mathbf{I}_s \subseteq \mathbf{I}_N$, and a set of χ FEC codes

$$F = \{(n_1, k_1, \mathbf{I}_{n_1}), (n_2, k_2, \mathbf{I}_{n_2}), \ldots, (n_\chi, k_\chi, \mathbf{I}_{n_\chi})\} \tag{11}$$

such that the following conditions are satisfied:

a) $s + c \leq \mathcal{R}_\mathcal{C}$, where $s = |\mathbf{I}_s|$;
b) $\mathbf{I}_{n_i} \subseteq \mathbf{I}_s$, for $i = 1, \ldots, \chi$;
c) $\sum_{i=1}^{\chi}(n_i - k_i) \leq c$;
d) the expected receiving distortion $E[D \mid F]$ is minimized.

In (11), one should note that in principle $\mathbf{I}_{n_i} \cap \mathbf{I}_{n_j} \neq \emptyset$, for $i \neq j$, as one data chunk may be protected by more than one FEC code. The index set $\mathbf{I}_s$ indicates the distribution of all s data chunks among the N multiresolution objects.

Suppose that there are L_κ chunks for each object κ. The expected receiving distortion $E[D \mid F]$ is given by

$$E[D \mid F] = \sum_{\kappa=1}^{N} \omega_\kappa \left(\sum_{l=0}^{L_\kappa} r(\kappa, l | F) D_\kappa(l) \right) \tag{12}$$

where $D_\kappa(l)$ is the resulting distortion from decoding the first l chunks of object κ, and $r(\kappa, l \mid F)$ is the probability that up to l chunks are recovered given the FEC codes F. Considering that a chunk is decodable if and only if all its preceding chunks in the same object are successfully decoded, we have

$$r(\kappa, l \mid F) = \begin{cases} \varepsilon(\kappa, l+1 \mid F) \prod_{i=1}^{l}[1 - \varepsilon(\kappa, i \mid F)], & l < L_\kappa \\ \prod_{i=1}^{L_\kappa}[1 - \varepsilon(\kappa, i \mid F)], & l = L_\kappa \end{cases} \tag{13}$$

where $\varepsilon(\kappa, i \mid F)$ is the error probability of chunk i in object κ, with given F and the chunk-erasure channel. For simplicity, we assume that the channel has an independent chunk loss process with a loss probability p. Then $\varepsilon(\kappa, i \mid F)$ can be easily calculated for given FEC codes [22].

To avoid overwhelming computation in finding a globally optimal solution, we solve the described problem by an iterative search consisting of two major steps. As the first step, we devise a parity-object generation algorithm to find, for a certain rate allocation $\Psi = \{s, c; \mathbf{I}_s\}$, a set of FEC codes that satisfies conditions (*b*–*d*). In the remaining step, a steepest decent search algorithm is performed, which finds the proper rate distribution between source and parity objects under the given rate constraint. The detailed solution is presented below.

A. Parity-Object Generation

To find the parity-object solution for given $\{s, c; \mathbf{I}_s\}$ that satisfies conditions (*b*–*d*), we develop a heuristic algorithm based on a fact revealed by Proposition 1. Namely, a joint FEC code is generally more efficient than separate FEC codes with the same proportion of redundancy. This fact inspires us to perform a search starting from a single parity object and gradually "split" the parity objects toward decreasing expected receiving distortion until an optimal point is reached.

Several notations are used in our presentation of the algorithm. 1) $\pi_{\kappa,l}$: *Importance* of chunk l in object κ, which is defined as the distortion of chunk l weighted by the object's weight factor and the chunk's loss probability. 2) Π: The sequence of s data chunks in a *decreasing* order of $\pi_{\kappa,l}$. 3) $f(\chi, h, \Pi)$: The FEC solution for a fixed number of χ parity objects, a number of h parity chunks, and a data chunk sequence Π.

Because decoding a chunk requires successful decoding of all the preceding chunks in mesh reconstruction, the distortion of chunk l is equivalent to the resulting distortion of decoding the first $l - 1$ chunks. Initially, the importance of chunk l in object

κ is assigned to be

$$\pi_{\kappa,l} = \omega_\kappa D_\kappa(l-1) \cdot p, \quad l = 1, 2, \ldots, L_\kappa \tag{14}$$

where p is the chunk-loss probability and $D_\kappa(l-1)$ denotes the resulting distortion from decoding the first $(l-1)$ chunks of object κ.

For a single parity object, i.e., $\chi = 1$, the FEC solution is given by $f(1, 0, \Pi) = \emptyset$, and

$$f(1, j, \Pi) = \mathop{\arg\min}_{(k+j,k,\mathbf{I}_k):k=1,2,\ldots,s} E[D \,|\, (k+j, k, \mathbf{I}_k)] \qquad j = 1, 2, \ldots, c \tag{15}$$

where $\mathbf{I}_k$ denots data chunks $0, 1, \ldots, k-1$ in sequence Π. Equation (15) implies that, for a single parity object with j parity chunks, the FEC solution protects the first k most important data chunks that result in the minimum expected distortion.

For a number of $\chi(\chi > 1)$ parity objects and h parity chunks, the algorithm recursively computes $f(\chi, h, \Pi), \chi \le h \le c$, by

$$\begin{cases} f(\chi, h, \Pi) = \arg\min_{F_j : j=1,2,\ldots,h-\chi+1} E[D|F_j] \\ F_j = f(\chi-1, h-j, \Pi) \cup f(1, j, \Pi') \end{cases} \tag{16}$$

where Π' denotes the reordered chunk sequence with updated chunk importance after applying FEC codes $f(\chi-1, h-j, \Pi)$. In other words, the FEC solution for $\chi > 1$ and a number of h parity chunks is the union of the FEC solution for $\chi - 1$ parity objects with $h - j$ parity chunks and the solution for 1 parity object with j parity chunks, with j selected such that a minimum expected distortion is obtained.

The algorithm then finds the optimal FEC solution by performing the following iteration:

$$F_{\text{opt}}^{(1)} = f(1, c, \Pi) \tag{17}$$

$$F_{\text{opt}}^{(\chi+1)} = \mathop{\arg\min}_{F \in \{F_{\text{opt}}^{(\chi)}, f(\chi+1, c, \Pi)\}} E[D \,|\, F] \tag{18}$$

$$\pi_{\kappa,l} = \omega_\kappa D_\kappa(l) \cdot \varepsilon\left(\kappa, l \,|\, F_{\text{opt}}^{(\chi+1)}\right) \tag{19}$$

until for a certain χ there exists

$$F_{\text{opt}}^{(\chi+1)} = F_{\text{opt}}^{(\chi)}.$$

Without much difficulty, the foregoing algorithm can be implemented using dynamic programming. It has a worst-case computation complexity of $O(sc) \cdot O(s \log s)$, accounting for the reordering operations needed in (16) for the chunk sequence. In practice, the algorithm reaches solutions in a faster computation time, as we expected from Proposition 1.

B. Rate Allocation

We use *ParityObject*$(\cdot)$ to denote the parity-object generation algorithm described above, which returns the optimal FEC solution F_Ψ and the corresponding expected distortion $E[D|F_\Psi]$ for a given rate distribution $\Psi = \{s, c; \mathbf{I}_s\}$. The optimal rate-allocation solution, Ψ_{opt}, under the rate constraint $\mathcal{R}_\mathcal{C}$ is then given by

$$\Psi_{\text{opt}} = \mathop{\arg\min}_{\Psi : s+c \le \mathcal{R}_\mathcal{C}} E[D \,|\, F_\Psi] \tag{20}$$

(i) Initialization: $m = 1, i_\kappa = 0, 1 \le \kappa \le N,$

$$s^{(1)} = \sum_{\kappa=1}^{N} R(M_\kappa^0), c^{(1)} = \mathcal{R}_\mathcal{C} - s^{(1)}; \mathbf{I}_s^{(1)} = \bigcup_{\kappa=1}^{N} \mathbf{I}(M_\kappa^{(1)}),$$

$$\{F^{(1)}, E[D|F^{(1)}]\} \leftarrow \textit{ParityObject}(s^{(1)}, c^{(1)}; \mathbf{I}_s^{(1)}).$$

(ii) Incremental allocation $(1 \le \kappa \le N)$*:*

$$\delta_\kappa = R(M_\kappa^{i_\kappa+1}),$$

$$s_\kappa = s^{(m)} + \delta_\kappa, c_\kappa = c^{(m)} - \delta_\kappa;$$

$$\mathbf{I}_{s,\kappa} = \mathbf{I}_s^{(m)} \cup \mathbf{I}(M_\kappa^{i_\kappa+1}),$$

$$\{F_\kappa, E[D|F_\kappa]\} \leftarrow \textit{ParityObject}(s_\kappa, c_\kappa; \mathbf{I}_{s,\kappa}).$$

(iii) Steepest decent search:

$$\kappa_{opt} = \mathop{\arg\max}_{\kappa=1,2,\cdots,N} \left(E[D|F^{(m)}] - E[D|F_\kappa]\right)/\delta_\kappa,$$

$$i_{\kappa_{opt}} = i_{\kappa_{opt}} + 1,$$

$$\{E[D|F^{(m+1)}], F^{(m+1)}\} = \{E[D|F_{\kappa_{opt}}], F_{\kappa_{opt}}\},$$

$$\{s^{(m+1)}, c^{(m+1)}; \mathbf{I}_s^{(m+1)}\} = \{s_{\kappa_{opt}}, c_{\kappa_{opt}}; \mathbf{I}_{s,\kappa_{opt}}\}.$$

(iv) Stopping condition:

If $E[D|F^{(m+1)}] > E[D|F^{(m)}]$, *Stop,*

Otherwise, $m = m + 1$, *Goto (ii).*

Fig. 5. Rate allocation using the parity-object generation algorithm, where $R(\cdot)$ denotes the bit rate of a particular LOD of an object measured in chunks and $\mathbf{I}(\cdot)$ gives the indices of corresponding data chunks in the entire sequence.

The solution to (20) is found by a steepest decent search algorithm starting from the lowest resolution for each object. The steepest decent search is performed in an incremental manner. At each step, the resolution of either one of the N objects is increased; the remaining bit rate is allocated to parity redundancy, and the corresponding optimal FEC solution is computed using the parity-object generation algorithm. Among N possibilities, the one that results in the maximum ratio of expected distortion reduction over rate increment is selected. The resolution of the corresponding object is then increased. This procedure is repeated until no distortion reduction is attained by any of the N possibilities. We summarize the major steps of the rate-allocation algorithm in Fig. 5.

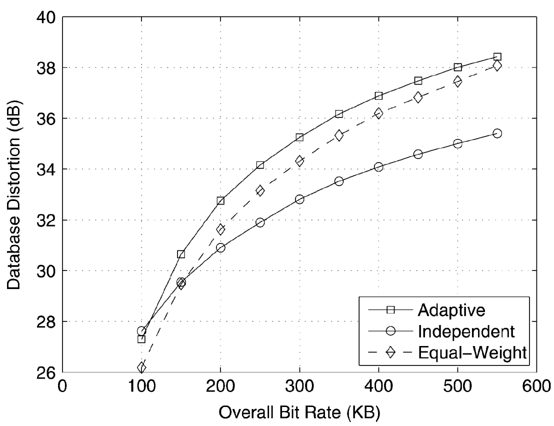
One special case of the rate-allocation in (20) is $c \equiv 0$, which corresponds to scenarios where there is no data loss or the rate constraint is solely imposed on the source coding. In such cases, parity object is not generated, and the rate-allocation algorithm returns to be distributing bits among all source objects toward the best distortion-rate performance.

VI. Experimental Results

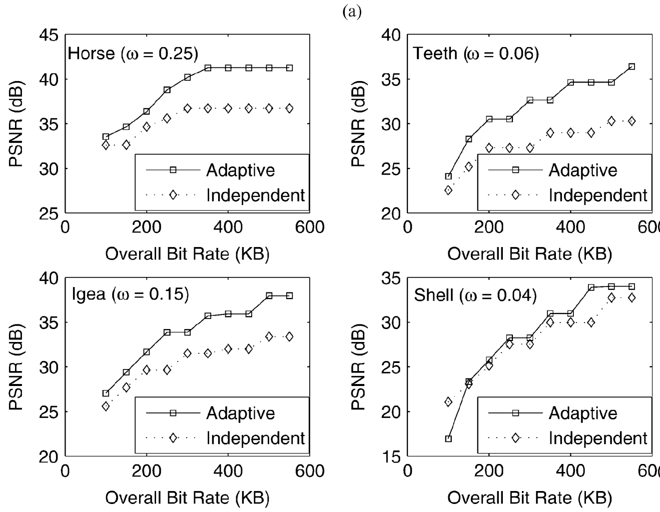
In this section, we present experimental results for evaluating the performance of the proposed source and channel coding system. A 3-D database containing ten mesh models (courtesy of Cyberware, Inc.) is used in our tests, which have preassigned weight factors ranging from 0.02 to 0.25 as listed in Table I. Each model is progressively simplified to generate various

TABLE I
TEST 3-D DATABASE

Model	# Faces	ω_κ	Model	# Faces	ω_κ
EYEBALL	39,600	0.05	HORSE	39,698	0.25
SCREWDRIVER	54,300	0.05	DINOSAUR	56,192	0.02
TEETH	58,300	0.06	IGEA	67,170	0.15
BALL JOINT	68,530	0.10	SANTA	75,778	0.20
ISIS	93,820	0.08	SHELL	97,928	0.04



(a)



(b)

Fig. 6. Distortion-rate performance of different coding schemes in an error-free environment: (a) distortion of the database; (b) distortion of selected objects.

LODs, and the compressed LOD hierarchy is packetized with a chunk size of 500 bytes. We use 13 bits for vector quantization.

As the first study, we consider an error-free transmission environment and investigate the distortion-rate performance of the proposed coding system compared with a conventional method that generates the VQ codebook using a training model set independent from the test database. This independent training set has a comparable size with the test database and is generated using the same parameters in the mesh simplification process.

Fig. 6(a) presents the obtained distortion-rate performance for the two comparing methods (the solid lines), where the x axis is the overall bit-rate constraint for all the objects and the y axis gives the PSNR value. The overall rate budget here does not include the bit rate of base meshes, which are compressed by the same algorithm in all coding schemes. The overall bit rate of adaptive VQ includes the bit rate of the VQ codebook, as it is required to be transmitted with the compressed bitstream. We calculate PSNR by

$$\mathrm{PSNR} = -20\log_{10}\mathcal{D}\,(\mathrm{dB}) \tag{21}$$

with $\mathcal{D}$ given by (2) and (3).

In Fig. 6(a), it is shown that, except for a very low bit rate ($\mathcal{R}_\mathcal{C} \leq 100$ KB), the distortion-rate performance of the scene database is significantly improved by using adaptive vector quantization. The exception is because the codebook overhead becomes unaffordable to adaptive VQ at the very low bit rate. At an overall rate of 300 KB, for example, adaptive VQ increases the quality of the database by 2.5 dB compared to independent VQ. The performance difference between the two schemes becomes larger as the bit rate increases, due to the error propagation while the database reconstruction proceeds to higher LODs. Respectively, Fig. 6(b) plots the distortion of several objects in the database for an individual comparison. As can be seen in the plots, although both schemes effectively distribute bit rates among the objects according to their weights, adaptive VQ greatly outperforms its independent counterpart. For the objects with higher weights, e.g., "horse" and "Igea," adaptive VQ quickly reaches a close-to-full resolution as the overall bit rate increases. Subjectively, this significant quality difference is confirmed by visually comparing the two pairs of models shown in Fig. 7(a)–(d), where the "horse" and "Igea" models decoded under the rate of 300 KB are captured.

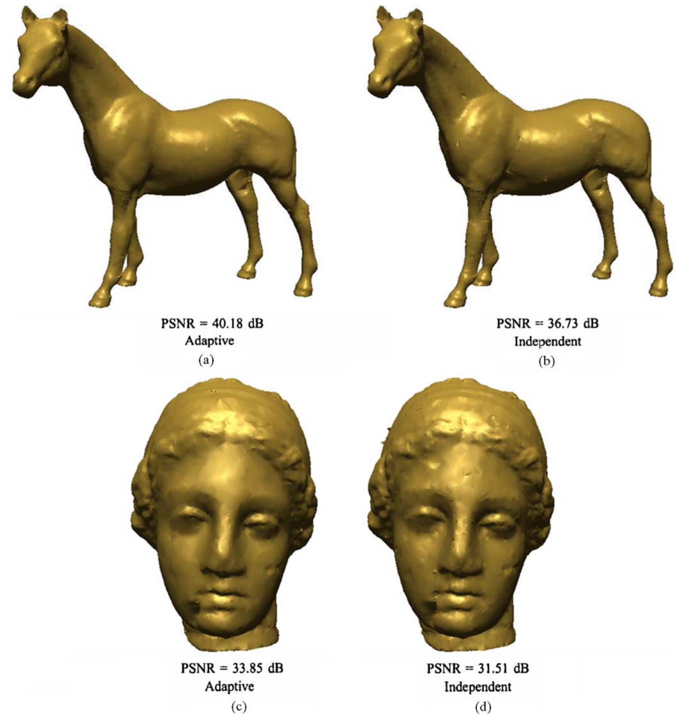


Fig. 7. Subjective comparison of decoded "horse" and "Igea" models under an overall bit rate of 300 KB. The relative bit rates allocated by adaptive VQ to the two models are 43 KB and 33 KB, respectively, after the codebook overhead is excluded: (a) $\mathrm{PSNR} = 40.18$ dB adaptive; (b) $\mathrm{PSNR} = 36.73$ dB independent; (c) $\mathrm{PSNR} = 33.85$ dB adaptive; (d) $\mathrm{PSNR} = 31.51$ dB independent.

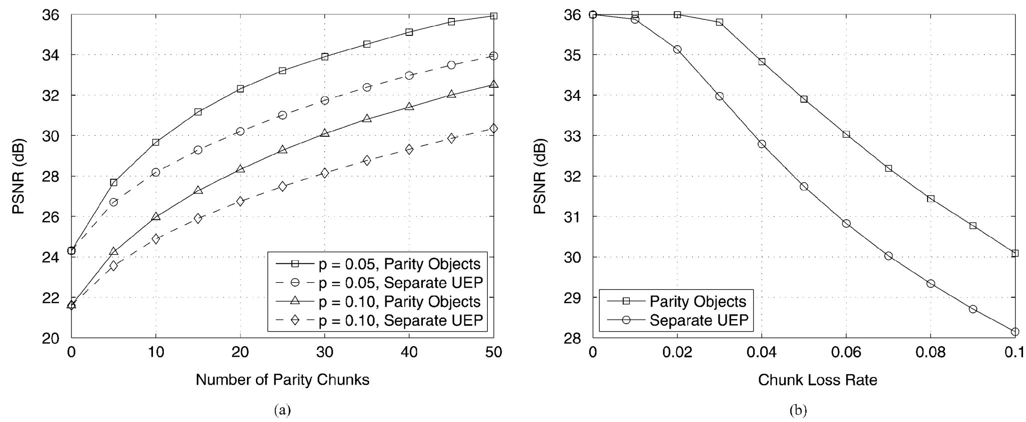


Fig. 8. Receiving distortion of the test database for the data-chunk distribution under 300 KB: (a) distortion versus parity redundancy for $p = 0.05$ and $p = 0.10$; (b) distortion versus chunk loss rates with 5% parity redundancy.

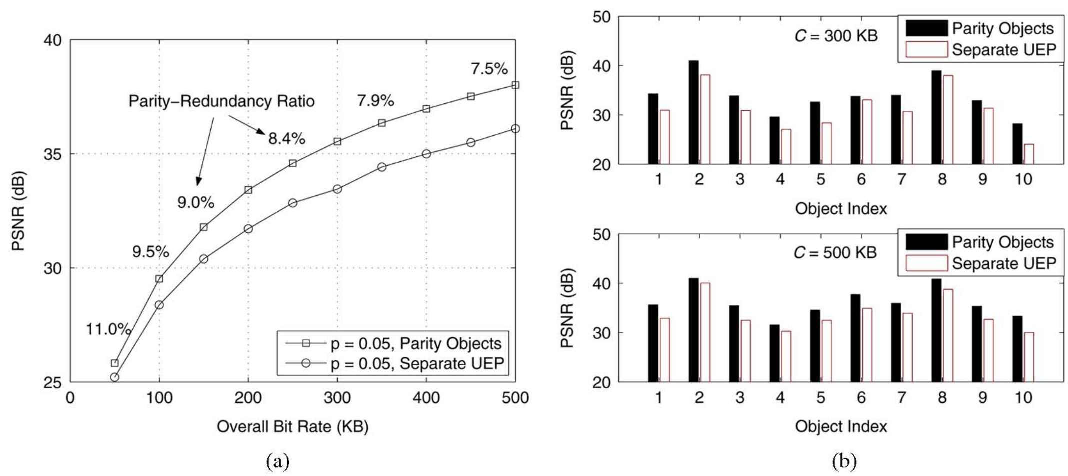


Fig. 9. Receiving distortion of the test database with rate allocation between source and parity objects under a chunk loss rate $p = 0.05$: (a) overall distortion; (b) individual cases under $\mathcal{C} = 300$ KB and $\mathcal{C} = 500$ KB. Objects are indexed in accordance with Table I from left to right, top down.

As a reference, in Fig. 6(a) we also provide the distortion-rate result obtained from a method that treats multiple objects and their various LODs equally importantly (the dashed line). Same as the proposed scheme, the equal-weight method uses prediction residuals from all mesh objects in the input database as training vectors. Nevertheless, the VQ codebook is trained by the original SR algorithm instead of the weighted training, and equal weights are deployed in the rate allocation for all the objects under given rate constraints. It is not surprising that adaptive VQ consistently outperforms the equal-weight method, due to the preferential rate allocation. More importantly, one should note that adaptive VQ achieves higher gains at low bit-rate ranges. This improvement comes from the weighted codebook training which allocates higher quantization precisions to higher-weight objects and lower LODs of each object, resulting in less error at the low bit rate.

To study the performance of the proposed coding system under a lossy environment, we compare our parity-object based transmission mechanism with the conventional object-oriented UEP scheme, which implements UEP for multiple objects respectively and appends parity data to the stream of each object. For a given parity budget, the algorithm incrementally distributes parity chunks among source objects until the maximum redundancy is reached. At each time, the assignment of one additional parity chunk to either of the multiple objects is determined by minimizing the expected receiving distortion. We refer to this comparing mechanism as *separate UEP* in our presentation.

We conduct the performance evaluation following the two stages presented in Section V. In the first stage, we apply the parity-generation algorithm for a fixed source-rate allocation obtained under 300 KB but with different budgets on parity redundancies or various chunk loss rates. The results of averaged receiving distortion are presented in Fig. 8(a)–(b). In particular, Fig. 8(a) plots the averaged receiving distortion of the two UEP schemes with gradually increased parity redundancy budgets under chunk loss rates $p = 0.05$ and $p = 0.10$, while Fig. 8(b) shows the receiving distortion for various loss rates under a fixed parity redundancy ratio of 5%. As we anticipate, the parity-object method stably outperforms separate UEP under all conditions. An approximately 2-dB improvement is quickly reached as the chunk loss rate and/or the parity redundancy ratio increase.

We then place overall bit-rate constraints on source and channel coding and perform joint rate allocation following the algorithm described in Fig. 5. Fig. 9(a) presents the receiving distortion of the scene database under different rate constraints. Similar to what we have perceived in the preceding results, using parity objects improves the receiving distortion by roughly 2 dB compared to separate UEP. The performance difference between the two comparing schemes reaches a stable level with an overall rate $\mathcal{C} \geq 300$ KB. In Fig. 9(a), the proportion of bit rate allocated to parity under the corresponding rate constraint is also marked. It is shown that, for a certain chunk loss rate, the parity-allocation ratio gradually decreases and converges to a constant as the overall bit rate increases. Fig. 9(b) provides, for two individual cases: $\mathcal{C} = 300$ KB and $\mathcal{C} = 500$ KB, the distortion of each mesh object. The results confirm the effectiveness of both schemes in protecting objects with higher weights, whereas the proposed joint coding method provides more efficient error recovery for the transmitted data chunks, hence resulting in higher receiving quality for every object in the database.

VII. Conclusion

The joint source and channel coding system proposed in this paper addresses the unique properties of 3-D scene databases in three components: adaptive vector quantization, partially ordered packetization, and joint unequal error protection (UEP). Modeling the unequal decoding importance of mesh objects in a scene database, the proposed coding system provides preferential treatment amongst multiple objects and various LODs of each object while preserving their decoding independencies in packet delivery. Operational algorithms were developed to realize the joint UEP under a rate-distortion optimization framework, and simulation results confirmed their efficacy. It is worth to mention that, although we exploited adaptive vector quantization to provide preferential treatment in coding the 3-D scene database, the parity-object based error protection and the rate allocation algorithm do not depend on the particular compression scheme but are applicable for general multiresolution hierarchies.

The adaptive vector quantization generates codebook using objects from the scene database. The objects from the database are not necessarily to have similar detail or connectivity, although it is anticipated that the similarity of objects may impact the compression, as it essentially reflects the entropy of the source data. In general, it is the application's responsibility to account for the geometric similarity of objects together with other factors to determine the coding scenarios. For a large scene database with various types of models, for example, the application may partition the entire database into subsets and code each subset separately using the proposed approach.

In this paper, we assumed predetermined and view irrelevant object weights for a given 3-D scene database. Although a view-dependent treatment may be introduced in the streaming mechanism, it is not considered in the presented work. Our future research will consider the interaction between specific object-weighting schemes and the source and channel coding, and develop streaming mechanisms that adapt to the rendering application. Also, the complexity and overhead of the proposed channel coding scheme will be further studied under real implementations, in which situations the optimal tradeoff between the coding and rendering complexities will be addressed.

References

[1] G. Taubin and J. Rossignac, "Geometry compression through topological surgery," *ACM Trans. Graph.*, vol. 17, no. 2, pp. 84–115, Apr. 1998.

[2] C. Touma and C. Gotsman, "Triangle mesh compression," presented at the Graphics Interface Conf., Vancouver, BC, Canada, Jun. 1998.

[3] H. Hoppe, "Progressive mesh," in *Proc. ACM SIGGRAPH*, 1996, pp. 99–108.

[4] G. Taubin, A. Gueziec, W. Horn, and F. Lazarus, "Progressive forest split compression," in *Proc. ACM SIGGRAPH*, 1998, pp. 123–132.

[5] J. Li and C.-C. Kuo, "Progressive coding of 3-D graphic models," *Proc. IEEE*, vol. 86, no. 6, pp. 1052–1063, Jun. 1998.

[6] R. Pajarola and J. Rossignac, "Compressed progressive meshes," *IEEE Trans. Vis. Comput. Graph.*, vol. 6, no. 1, pp. 79–93, Jan. 2000.

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

[8] E.-S. Lee and H.-S. Ko, "Vertex data compression for triangular meshes," *Pacific Graph.*, pp. 225–234, 2000.

[9] P. H. Chou and T. H. Meng, "Vertex data compression through vector quantization," *IEEE Trans. Vis. Comput. Graph.*, vol. 8, no. 4, pp. 373–382, Apr. 2002.

[10] J. Li, D. Tian, and G. AlRegib, "Vector quantization in multi-resolution mesh compression," *IEEE Signal Process. Lett.*, vol. 13, no. 10, pp. 616–619, Oct. 2006.

[11] A. Albanese, J. Blomer, J. Edmonds, M. Luby, and M. Sudan*, Priority Encoding Transmission*, vol. 42, pp. 1737–1744, 1996.

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

[13] 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. Sel. Areas Commun.*, vol. 18, no. 6, pp. 819–828, Jun. 2000.

[14] J. Kim, R. M. Mersereau, and Y. Altunbasak, "Error-resilient image and video transmission over the internet using unequal error protection," *IEEE Trans. Image Process.*, vol. 12, no. 2, pp. 121–131, Feb. 2003.

[15] G. Al-Regib, Y. Altunbasak, and R. M. Mersereau, "Bit allocation for joint source and channel coding of progressively compressed 3-D models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 256–268, Feb. 2005.

[16] G. AlRegib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively compressed 3-D models," *IEEE Trans. Multimedia*, vol. 7, no. 4, pp. 766–776, Dec. 2005.

[17] M. Marka and J. E. Fowler, "Unequal error protection of embedded multimedia objects for packet-erasure channels," in *Proc. Int. Workshop on Multimedia Signal Processing*, 2002, pp. 61–64.

[18] T. Gan and K.-K. Ma, "Weighted unequal error protection for transmitting scalable object-oriented images over packet-erasure networks," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 189–199, Feb. 2005.

[19] J. Thie and D. Taubman, "Optimal erasure protection assignment for scalable compressed data with small packets and short channel codewords," *EURASIP J. Appl. Signal Process.*, no. 2, pp. 207–219, 2001.

[20] P. Cignoni, C. Rocchini, and R. Scopigno, "Metro: Measuring error on simplified surfaces," in *Proc. Eurographics*, 1998, vol. 17, no. 2, pp. 167–174.

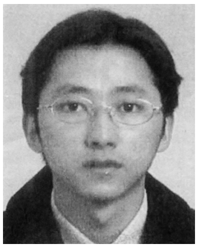
[21] K. Zeger, J. Vaisey, and A. Gersho, "Globally optimal vector quantizer design by stochastic relaxation," *IEEE Trans. Signal Process.*, vol. 40, no. 2, pp. 310–322, Feb. 1992.

[22] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 27, no. 2, pp. 24–36, 1997.

Dihong Tian (M'07) received the B.S. degree in computer science and technology and the M.S. degree in electronic engineering and information science from the University of Science and Technology of China (USTC) in 1999 and 2002, respectively, and the Ph.D. degree in electrical and computer engineering in 2006 from the Georgia Institute of Technology (Georgia Tech), Atlanta. His Ph.D. research was devoted to multimedia communications and networking, particularly on streaming video and 3-D graphics.

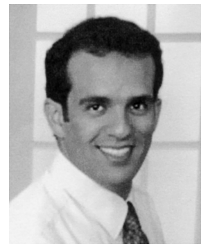
Since 2006, he has been with Cisco Systems, Inc., San Jose, CA, working on video coding and processing for TelePresence systems.

Dr. Tian is an alumnus of the Special Class for the Gifted Young (SCGY) at USTC and an alumnus of the Center for Signal and Image Processing (CSIP) at Georgia Tech. He was the recipient of the Guo Mo-Ruo Award at USTC and the CSIP Outstanding Research Award at Georgia Tech.



Junlin Li (S'05) received the B.S. and M.S. degrees in electronic engineering from Tsinghua university, Beijing, China, in 2001 and 2004, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

He spent the summer of 2006 as an Research Intern in the Microsoft Research (MSR). His research interests include multimedia signal processing, distributed signal processing, and wireless sensor networks.



Ghassan AlRegib (M'07) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 2003.

He is an Assistant Professor in the School of Electrical and Computer Engineering, Georgia Tech. He joined the faculty at Georgia Tech in August 2003, where his research group is currently working on projects related to multimedia processing and communications, distributed processing, collaborative systems, and wireless sensor networks.

Dr. AlRegib was the recipient of the Outstanding Graduate Teaching Award in 2000-2001 at the School of Electrical and Computer Engineering at Georgia Tech, the Center for Signal and Image Processing Research Award in Spring 2003 at Georgia Tech, and the Center for Signal and Image Processing Service Award in Spring 2003 at Georgia Tech. He is the Co-Chair of theFirst International Conference on Immersive Telecommunications (IMMERSCOM 2007) to be held in Italy in October 2007. He is also the Chair of the Special Sessions Program at the IEEE International Conference on Image Processing (ICIP) 2006 in Atlanta. He also serves as an Associate Editor of the *IEEE Signal Processing Magazine*