2009 Eighth IEEE International Symposium on Network Computing and Applications

Unequal Error Protection (UEP) for Wavelet-Based Wireless 3D Mesh Transmission

Fan Wu
Computer Science Department
Tuskegee University
Tuskegee, Alabama 36088
Email: wuf@tuskegee.edu

Emmanuel Agu, Clifford Lindsay
Computer Science Department
Worcester Polytechnic Institute
Worcester, Massachusetts 01609
Email: {emmanuel,clindsay}@cs.wpi.edu

Chung-han Chen
Computer Science Department
Tuskegee University
Tuskegee, Alabama 36088
Email: jchen@tuskegee.edu

Abstract— **The recent popularity of networked graphics applications such as distributed military simulators and online games, has increased the need to transmit large 3D meshes and textures over wireless networks. To speed up large mesh transmission over low-bandwidth wireless links, we use a wavelet-based technique that aggressively compresses large meshes and enables progressive (piece-wise) transmission. Using wavelets, a server only needs to send the full connectivity information of a small base mesh along with wavelet coefficients that refine it, saving memory and bandwidth. To mitigate packet losses caused by high wireless error rates, we propose a novel Forward Error Correction (FEC) scheme based on Unequal Error Protection (UEP). UEP adds more error correction bits to regions of the mesh that have more details. Our work uses UEP to make wavelet-encoded meshes more resilient to wireless errors. Experimental results shows that our proposed UEP scheme is more error-resilient than No Error Protection (NEP) and Equal Error Protection (EEP) as the packet loss rate increases by achieving 50% less relative errors and maintaining the decoded mesh structure. Our scheme can be integrated into future mobile devices and shall be useful in application areas such as military simulators on mobile devices.**

Index Terms- Wavelets; Multiresolution Rendering; Ubiquitous Graphics; Error Protection.

I. Introduction

Networked graphics applications such as multiplayer online games, 3D maps and distributed military simulations have become popular, increasing the need to transmit 3D graphics content from storage servers, over networks to mobile devices. The Internet is becoming a communication medium into a virtual reality world where users can interact with each other. Networked 3D graphics applications use highly detailed 3D models, large meshes and textures that need to be stored, transmitted, and rendered efficiently, especially when available bandwidth is scarce.

To mitigate the widely variable network bandwidths of heterogeneous networks and the wide range of resources available on mobile devices for graphics rendering, we proposed UbiWave [1], a system for scalable rendering on heterogeneous computing devices. UbiWave encompasses all stages including compact storage of graphics content as wavelets, efficient transmission and rendering on the mobile device. This wavelet-based framework enables small mobile devices such as cell phones to download and render extremely large captured content that can be many gigabytes in size. Figure 1 is an overview of our proposed approach. Large graphics content is encoded using wavelets (on the left of figure 1). When retrieved, the content is dynamically scaled to the resources of a mobile device and wireless network, transmitted wirelessly to the mobile device where it is rendered (on the right of figure 1). The realism of rendering on the mobile device is varied to accommodate mobile device constraints on screen size and battery power. Essentially, small devices such as cell phones on a GPRS cellular data network and laptop on a broadband WiMax network, can download and render the same source scenes, but automatically achieve different resolutions that are optimal for their configurations.

The UbiWave framework can be used in many 3D mobile graphics applications, such as mobile gaming, mobile 3D maps and so on. UbiWave can be used as a stand-alone software tool that downloaders of large scanned content can use offline. In a more ambitious scenario, the quality of rendered images in mobile graphics applications would be varied dynamically based on available resources. For instance, the geometry of rendered objects and the quality of shading of a mobile flight simulator could be gracefully degraded as the device's battery dies.

Several compression [2]–[5] techniques have been developed to minimize transmission times on low-bandwidth network links by reducing transmitted mesh sizes. Additionally, the wireless channel is well known to have significantly high error rates. Retransmission of damaged packets or Forward Error Correction (FEC) are two strategies that are frequently used to mitigate wireless channel errors. However, the roundtrip delays caused by retransmissions in network protocols such as TCP/IP and the IEEE 802.11 Wireless LAN protocol is experienced as latency to users, which can affect the interactivity of networked graphics applications. For such interactive applications, FEC is preferred to retransmissions as an error mitigation strategy. FEC schemes add redundant bits to the original meshes before transmission such that minor errors can be corrected by the receiver, thus avoiding retransmissions.

978-0-7695-3698-9/09 $25.00 © 2009 IEEE
DOI 10.1109/NCA.2009.40

IEEE computer society

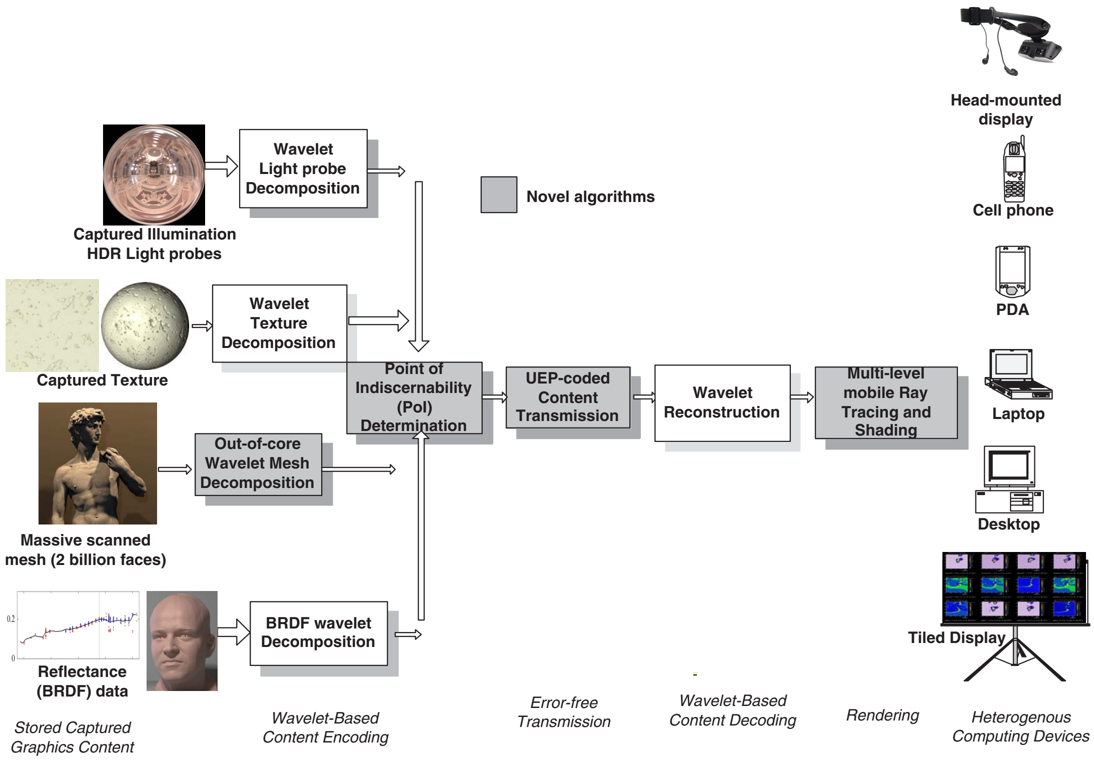


Fig. 1. UbiWave: Our Wavelets-based Ubiquitous Graphics System

This paper focusses on the mitigation of transmission errors in UbiWave during the transmission of 3D meshes to mobile hosts. To speed up large mesh transmission over low-bandwidth wireless links, we encode meshes and all graphics content using wavelets prior to transmission because wavelets achieve over 100x compression ratios and facilitates piece-wise transmission of large meshes. Using wavelets, a server only needs to send a small base mesh along with wavelet coefficients that refine it, saving memory and bandwidth.

As our main contribution in this paper, we propose a FEC scheme to protect wavelet-encoded meshes from wireless errors. The Hamming code [6] and Reed-Solomon [7] codes are two popular FEC schemes that mitigate error well for most applications. However, FEC schemes that consider the underlying structure of wavelet-encoded content frequently outperform more general schemes that do not. Wavelet-specific FEC techniques for image [8] and video transmission [9] have been proposed, but not for wavelet-encoded meshes. Our proposed FEC scheme is based on the principle of Unequal Error Protection (UEP). In UEP [10], the number of FEC bits alloted to each part of the mesh is proportional to the amount of information it contains: more bits are added to parts with more information. Thus, areas of a mesh such as a human face that has many fine details are allocated more FEC bits than areas such as the back with less details. Previous work applied UEP to Compressed Progressive Meshes (CPM). Our work uses UEP to make wavelet-encoded meshes more resilient to wireless errors.

The rest of the paper is as follows. Section II discusses related work, Section III provides some background on wavelets and UEP, Section IV describes our UEP scheme for wavelet-encoded meshes, Section V describes our channel model and simulation results, and Section VI presents our conclusions and future work.

II. Related Work

Recent research in the transmission of 3D content over unreliable links have focussed mostly on still images and video sequences [11]. The compression of meshes is another active area of research. Very little research has addressed the issue of transmitting 3D graphics models over wireless networks. This is partly due to the fact that popular applications such as multiplayer games, which require this service have only recently emerged. Existing techniques for mitigating error while transmitting graphics models include robust error coding to retransmission schemes for damaged network packets.

A retransmission-based error-resilient technique has been proposed by Bischoff and Kobbelt [12]. In their scheme, the base mesh is re-transmitted along with every

Level-of-Detail (LoD) to guarantee that it is correctly received at the mobile client. However, the overhead of transmitting the base mesh can be significant, making this scheme inefficient when packet loss rate is low.

Bajaj *et al* [13] proposed several robust source coding methods for meshes. Even though this method adds a level of protection to the transmitted mesh, it does not adapt well to different ranges of channel packet loss rate. Yan *et al* [14] propose partitioning a 3D model into several segments that are then transmitted independently. However, they use experimental calibration to determine the number of error-protection bits assigned to different segments before transmission, which can be time-consuming. Our proposed technique applies an analytic distortion metric to determine the number of bits assigned per segment and does not require experimental calibration. MPEG-4 also uses error-resilient coding of 3D models that is similar to that proposed by Yan *et al* [14]. UEP is an error coding paradigm that assigns FEC bits based on the the amount of information a given segment contains. Al-Regib *et al* [10], [15] applies UEP to the Compressed Progressive Mesh (CPM) [16], a popular mesh representation in order to increase its resilience to transmission errors. As our main contribution, we apply UEP method to meshes that have been encoded using wavelets to make them more resilient to wireless errors. We note that UEP encoding of any content closely depends on a) the underlying structure of the content to be encoded and b) the ability to determine the relative importance of different parts of the mesh.

III. Background

A. Multiresolution analysis of meshes using wavelets

Overview: Wavelets are a mathematical tool that can *decompose* large input meshes to yield a coarse (rough) base mesh, plus a tree of detail coefficients, as shown in Figure 2. *Reconstructing* the original mesh starts from the coarse base mesh. Its resolution is then successively improved by adding more levels of the wavelet detail coefficient tree. We have previously proposed a system for ubiquitous graphics called UbiWave, in which all rendering inputs such as meshes [17], textures [18] and material reflectance properties [19] are encoded as wavelets prior to transmission. The ability of wavelets to present 3D graphics at different Levels-of-Detail facilitates fine-grained scalable rendering on small computing devices even when the original inputs are extremely large files. Wavelets have been used in a wide range of applications including graphics and image processing, information retrieval, FBI fingerprint storage and geographic modeling. Today, published work has shown that almost all aspects of a graphics scene can be decomposed using wavelets including meshes, textures, material and reflectance properties. If graphics content is available as decomposed wavelets, mobile devices can retrieve scaled down resolutions suitable for their use. Wavelets achieve aggressive compression (up to 100x), which reduces transmission time over low-bandwidth cellular networks. Wavelets also support progressive refinement such that users can view intermediate mesh resolutions as more pieces are retrieved during mesh download. Finally, using wavelets for graphics content facilitates integration of emerging mobile graphics standards with existing MPEG4 video and JPEG2000 image standards, where wavelets are already being used.

Wavelets in computer graphics: Schroeder [20] was one of the first to use wavelets in computer graphics and used wavelets to compress geometry and evaluate global illumination rendering equations. We focus on the application of wavelets to represent large input meshes at various levels of detail (also called multi-resolution analysis). In our mobile framework, our goal is to select the mesh Level-of-Detail (LoD) that is suitable for each mobile display's resolution. In general, we decompose the large mesh at a server, transmit a highly compressed file to the mobile device where a reconstruction procedure is used to regenerate the original mesh. During wavelet decomposition, a mesh is subdivided and deformed to make it fit the original surface to be approximated. Wavelet decomposition generates a base mesh, along with smoothness and detail wavelet coefficients. During reconstruction, starting with the base mesh, as more wavelet coefficients are included, a higher resolution mesh is generated. These reconstruction steps can be repeated until the desired mesh resolution (LoD) is achieved. The base mesh M^0, the original mesh M^j and all intermediate mesh LoDs form a multiresolution hierarchy of meshes. This process is shown in figure 2.

B. Unequal Error Protection

Approaches to mitigate wireless channel errors packets losses can be network-oriented solution such as retransmissions in TCP, post-processing solutions such as error concealment, or pre-processing solutions such as Forward Error Correction (FEC) codes. The roundtrip delay incurred make retransmissions unsuitable for interactive graphics applications. In multicast environments, retransmissions would also flood the sender with acknowledgements and performance could suffer. We consider the use of FEC. FEC strategies include Equal Error Protection (EEP) and Unequal Error Protection (UEP). EEP methods apply the same FEC code to all parts of the mesh's bit stream and is suitable when the channel has a low packet loss rate. However, at higher packet loss rates, considerable degradation on the decoded model quality may occur because of the high possibility that important parts might

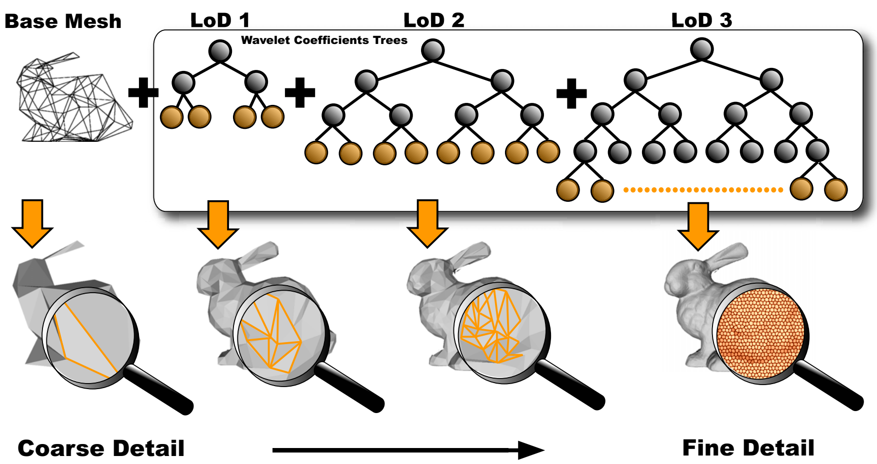


Fig. 2. Wavelets-based Multiresolutions

be lost. In this case, UEP is more suitable since important parts of decoded mesh get more assigned more FEC bits.

In this paper, after applying wavelets decomposition to a mesh, the base mesh as well as wavelet coefficients are assigned an FEC code rate depending on their contribution to the decoded mesh quality. The distribution of these FEC codes is calculated using a statistical distortion measure. Based on this measurement, we determine the number of error-protection codes to be assigned to the base mesh and each level of detail. The FEC codes used in this paper are Reed-Solomon (RS) codes. These error codes are perfect for error protection against bursty packet losses because they are maximum distance separable codes. An (n, k) RS-code encodes k information symbols where each symbol is represented by q bits. These k symbols are encoded into a codeword of n symbols, which is restricted by $n \leq 2^q - 1$. As soon as k symbols are received, all lost symbols can be reconstructed.

IV. Unequal Error Protection of Wavelet-Encoded Meshes

A. UEP in Wavelet-Based Multiresolution

After wavelet decomposition, the base mesh and first few levels of wavelet coefficient tree should be strongly protected to prevent packet loss. We examine several strategies for adding Forward Error Correction (FEC) bits to the base mesh and wavelet coefficients. First, we apply Equal Error Protection (EEP) where an equal number of FEC bits are applied to all parts of the base mesh and to all levels of the wavelet coefficient tree. That is, S^1 = S^2 = ... = S^{M+1}, where S^k is the number of FEC bits added to on the k^{th} level of wavelet coefficients. Next, we propose applying Unequal Error Protection (UEP) where bits in the encoded mesh are classified based on their contributions to the final look of the reconstructed mesh. Each class is then protected by a number of FEC bits that can provide a certain level of protection against channel losses. In our research, each level of the wavelet coefficient tree and the base mesh, is assigned an FEC code based on amount of distortion that would be introduced into the reconstructed mesh if that portion of the bitstream is lost. Parts of the bitstream that distort the look of the reconstructed mesh most when they are lost are the most important and hence we apply the largest portion of the FEC bit budget. Wavelet coefficients with large absolute values contain the most detail receive more error bit budget, since this level of coefficients contains more information (e.g. fine details such as eyes and nose of a face) compared to other levels. The FEC codes used are the Reed-Solomon (RS) codes. Reed-Solomon codes are block-based error correcting codes with a wide range of applications for error protection against burst packet losses. We also adapt our encoding order of our bitstream to further increase resilience to burst errors. The output bitstream is encoded in blocks of packets, where the data is placed in horizontal packets and then RS is applied across the block of packets vertically. Each block of packets is protected with a FEC code that is proportional to the importance of the corresponding base mesh or coefficients.

Since all types of error protection add extra bits to the original mesh bitstream prior to transmission, both EEP and UEP incur overheads that reduce the number of actual data bits sent compared with NEP. However, since reconstruction starts from the base mesh, loss of the base mesh or parts of it are particularly devastating. Essentially, the base mesh as well as coarser wavelet coefficients are more important than detail coefficients. At high packet loss rates, losing the base mesh or coarser wavelet coefficients degrades the decoded mesh quality significantly even if the detail coefficients are received correctly. EEP distributes error correction bits equally to the base mesh, and all levels of detail coefficients.

B. Distortion Measure

To determine the level of channel coding associated with each level of the wavelet coefficient tree, we need to evaluate the importance of those coefficients. In this section, we develop a distortion metric that evaluates the relative importance of the various levels of a wavelet coefficient tree. After we determine the importance of each level of the wavelet coefficient tree, we can then assign a fraction of the total FEC bits that is proportional to their importance. The main factors integrated into this distortion measure are: 1) The amount of information contained in the wavelet coefficient, 2)the total number of error-protection bits. As figure 3 shows, in each LoD, some new coefficients are added to the mesh, which provide more detailed information to the final rendered mesh.

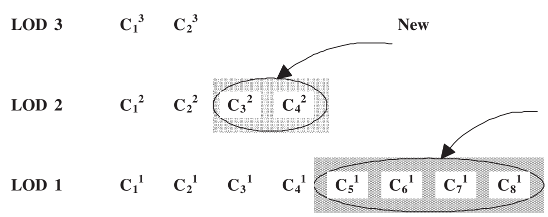


Fig. 3. Wavelet coefficient tree for a mesh with three LODs. C_i^j is the wavelet coefficient at level j.

To calculate the importance of each level of the wavelet coefficient tree, we evaluate the distortion that would be present in the final decoded mesh if all the coefficients in that level of the tree were lost. We associate a coefficients distortion quantity, $D_{wLOD}^{(j)}$ with the j^{th} LOD, which is defined as the average distortion (per coefficient) added when all coefficients that are added by this LOD are lost. The $D_{wLOD}^{(j)}$ is given by:

$$D_{wLOD}^{(j)} = \frac{1}{N_j} \sum_{i=1}^{N_j} |c_i^j| \quad (1)$$

Where N_j is the number of coefficients added by $LOD^{(j)}$. This distortion measure estimates the error between the meshes with the j^{th} LOD and the $(j+1)^{th}$ LOD. We use this distortion measure to calculate the fraction of the total error protection bit budget that is assigned to each level in UEP. In EEP, the available error protection bit-budget can be calculated as follows:

$$S = \sum_{j=1}^{M+1} (n-k) \times q \times B^j \quad (2)$$

where q is the codeword size. B^j is the number of codewords in each horizontal packet. In the case of UEP, the bit-budget, S, and the total packet size, n, are provided. Therefore, the RS code rates for all M layers need to be computed. Let α^j be the portion of the total bit-budget to protect j^{th} level of decoded mesh. That is, $\alpha^j = \frac{S^j}{S}$. So the j^{th} level bit-budget is given by:

$$(n-k_j) = \frac{\alpha^j \times S}{q \times B^j} \quad (3)$$

From Equation 3, we know α^j is the main factor to determine the RS code rate. We set α^j to be equal to the coefficients distortion quantity, $D_{wLOD}^{(j)}$ which was given in Equation 1. In this way, we can calculate RS code $(n-k_j)$ using Equation 3 for each part of decoded mesh.

C. Block-based Encoding

To further increase the error-resilience of our transmitted meshes, we apply block-based encoding after UEP encoding, before transmission. A simple example of our approach to block-based error correcting is described. Consider a 3D model that has been decomposed into a base mesh and three levels of wavelet coefficients (L_1, L_2 and L_3). Applying RS codes, the resulting packets are shown in Figure 4. The base mesh consists of five data packets with five error protection packets. The wavelet coefficients corresponding to level one, L_1, consists of six data packets with four error protection packets. Wavelet coefficient level L_2 consists of eight data packets with two error protection packets and level L_3 consists of ten data packets with no error protection packets. The base mesh and its associated RS packets are transmitted first, followed by the coarse wavelet coefficients, until the finest one. As shown in Figure 4, more FEC codes are assigned to the coarser level of coefficients than the finer one. Such an allocation of FEC codes is calculated by a distortion quantity that is described above. At a certain packet loss rate, some of the packets will be lost. Taking an example of three packets for each block being lost. Since the base mesh uses (10,5) error correction codes, when the number of lost packets is not more than five, the client can recover all lost packets. Therefore, in this example, it can recover all three lost packets. For the same reason, all three lost packets in L_1 can be recovered. But the lost packets in L_2 and L_3 can not be recovered by assigned RS codes. At client, the base mesh and L_1 level of coefficients have adequate protection but L_2 and L_3 levels of coefficients get lost. Therefore, the more important parts of the mesh are protected and correctly received by the client and decoded even when wireless channel loses a significant number of packets.

V. Result

In this section, we describe tests that we conducted using meshes to evaluate the performance of our method. In particular, the performance of the UEP, EEP and NEP

Data Transmission →

Base Mesh	1	2	3	4	5	RS	RS	RS	RS	RS
L_1	1	2	3	4	5	6	RS	RS	RS	RS
L_2	1	2	3	4	5	6	7	8	RS	RS
L_3	1	2	3	4	5	6	7	8	9	10

Fig. 4. Example of transmitted packets in UEP methods

are compared. First we describe a two-state Markov model known as the G-E model [21] for the wireless channel.

A. Channel Model

We use a Markov model with only two states to model a wireless channel with high bit error rates [21]. We shall now briefly describe its main characteristics.

G-E models are defined by the distribution of error-free intervals, which are called gaps. The gap is defined as the interval of length $v-1$ packets between two consecutive received error packets. This model is illustrated in figure 5 and the probability density function (pdf) $g(v)$ and cumulative distribution function (cdf) of the gaps greater than $v-1$ packets $G(v)$ are defined as equation 4 and equation 5, respectively.

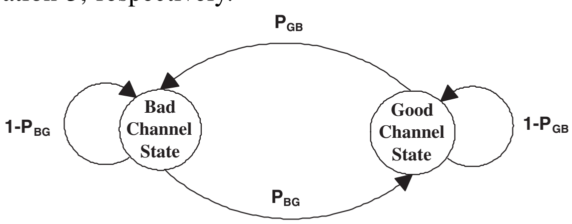


Fig. 5. G-E two state Markovian Channel Model. P_{GB} is the transition probability from the good state to the bad state while P_{BG} is the transition probability from the bad state to the good state

$$g(v) = \begin{cases} 1 - P_{BG} & , \quad v = 1 \\ P_{BG}(1-P_{GB})^{v-2} P_{GB} & , \quad v > 1 \end{cases} \tag{4}$$

$$G(v) = \begin{cases} 1 & , \quad v = 1 \\ P_{BG}(1-P_{GB})^{v-2} & , \quad v > 1 \end{cases} \tag{5}$$

Let $R(m,n)$ denote the probability of having $m-1$ packet losses within the $n-1$ packets following a lost packet. Then $R(m,n)$ is given by:

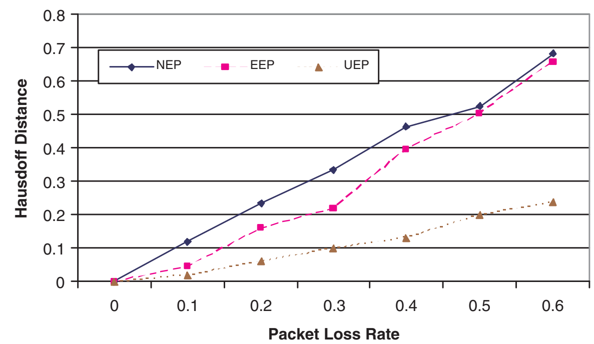
$$R(m,n) = \begin{cases} G(n) & , \quad m = 1 \\ \sum_{v=1}^{n-m+1} g(v) R(m-1, n-v) & , \quad 2 \le m \le n \end{cases} \tag{6}$$

So, the probability of losing m symbols, each of which is of q bits in length, within a block of n symbols is:

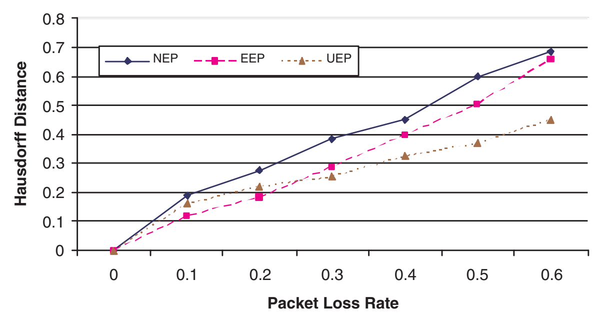
$$p(m,n) = \begin{cases} \sum_{v=1}^{n-m+1} P_B g(v) R(m, n-v+1) & , \quad 1 \le m \le n \\ 1 - \sum_{m=1}^{n} p(m,n) & , \quad m = 0 \end{cases} \tag{7}$$

B. Simulation Results

We applied the proposed unequal error protection (UEP) method on several models and here we report the results for the small mesh. We consider three cases: encoding the original mesh into a base mesh and 5 levels of detail, 10 levels of detail and 15 levels of detail. In general, the more levels of detail we use, the less information each layer contains. We use the Hausdorff distance to measure the amount of distortion in our received mesh. The Hausdorff distance expresses the geometric distance between two surfaces as the maximum of all pointwise distances. In general, more distortion increases the Hausdorff distance.



a



b

Fig. 6. Maximum Error(Hausdorff distance) between transmitted and decoded mesh when the RS code used for EEP is a: (n,k) = (63,45) and b: (n,k) = (63,51). NEP: no error protection is applied, EEP: equal error protection is applied, and UEP: unequal error protection is applied

Figure 6 depicts the distortion as a function of the packet loss rate for small mesh model. Three curves in this figure represent the cases of EEP, UEP, and NEP with level 5. As seen from these curves, for an error-free channel no packets are lost and the distortion in the transmitted mesh is zero. As the packet loss rate increases, the performance of EEP and NEP become closer to each other since neither technique can recover when packets of the base mesh or coarse level of coefficients are lost. However, UEP manages to protect the base mesh and coarse wavelet coefficients by assigning more error-protection bits and therefore improving the quality of the decoded mesh is better compared to other two methods. When the packet loss rate $P_{LR} \geq 0.2$, the base mesh information is lost and only UEP is able to protect the base mesh.

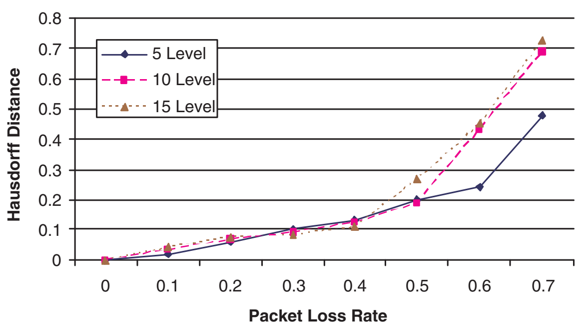


Fig. 7. Maximum Error(Hausdorff distance) between the transmitted and the decoded mesh when different level of detail (5,10,15) are used with RS code (n, k) = (63,45)

Figure 7 shows the distortion as a function of the packet loss rate for small mesh. Three curves in this figure represent the cases of 5, 10, 15 levels of detail. The figure shows a slow increase in the Hausdorff distance up till a knee point at which the Hausdorff distance (or distortion) increases quickly. Before the knee point, only wavelet coefficients are lost while the base mesh is correctly received. Beyond the knee points the high error rates cause the base mesh to get lost, causing a large increase in distortion (Hausdorff distance). The knee point of 5-level LoD is larger (more resilient to errors) than that of 10-level and 15-level LoDs. This is intuitive since as the mesh is encoded into more LoD levels, each level of the wavelet coefficient tree as well as the base mesh all receive fewer error protection bits. Hence, meshes that are encoded into more LoD levels will lose base mesh information easier than meshes encoded with fewer LoD levels. Thus for a fixed UEP bit budget, we find an inverse relationship between the number of mesh LoDs used and the error resilience of wavelet-encoded mesh. Before knee points, the base mesh is received and only wavelet coefficients are lost. As the mesh is encoded into more LoDs, the importance of each level of wavelet tree level is reduced and the degradation introduced when wavelet coefficients are lost is also reduced. Therefore, before knee point, the distortion of meshes encoded with more LoDs is slightly lower than that of meshes that use fewer LoDs.

Objective results have been presented above. We also compare the three methods, NEP, EEP, and UEP, subjectively by looking at images of the final rendered mesh after passing them through a simulated wireless channel. Figure 8 shows the experimental results for the small mesh. The first column on the left shows the decoded mesh in the NEP case for different packet loss rates. Similarly, the second and the third columns show the decoded meshes for EEP and UEP respectively. As shown, UEP maintains a reasonable decoded mesh quality as the packet loss rate increases. We have encoded the mesh into 5 Levels of Detail. As the error rate increases, UEP loses some detail coefficients but the base mesh and coarse coefficients are adequately protected and correctly received. Hence, only minor artifacts can be observed on the UEP as error rates increase. We can thus conclude that using our proposed UEP method on wavelet multiresolution, the quality of the decoded meshes is better as the packet loss rate increases.

VI. Conclusion and Future Work

We have presented Unequal Error Protection (UEP), a Forward Error Correction (FEC) scheme for the error-resilient transmission of meshes that have been encoded using wavelets, to increase decoded mesh quality. Error-protection bits are allocated according to the importance of parts of the wavelet-encoded mesh. The importance of each level is determined by a distortion measure that reflects the information the coefficients contain. Theoretically, the UEP method increases the resilience of wavelet-based mesh transmission to high error rates. By simulating mesh transmission using our proposed scheme on two different channel models, we compare the performance of the proposed UEP, EEP and NEP methods.

In future work, we shall also compare the performance of the proposed UEP scheme when applied to wavelet-encoded meshes to UEP on Compressed Progressive Meshes. We would also like to investigate the benefits of *zero-tree coding*. In zero tree coding, when wavelets coefficients are encoded, in each level, coefficients with values greater than some appropriate threshold value are kept and low-valued coefficients (little information) are replaced by zero. Finally, we would like to use Hoffman coding to further reduce the number of bits transmitted.

References

[1] F. Wu, E. Agu and M. Ward, "Multiresolution Graphics on Ubiquitous Displays using Wavelets", *International Journal of Virtual Reality, Vol. 5, No. 3*, 2006.

[2] C. Touma and C. Gotsman, "Triangle mesh compression," *Proc. Graphics Interface, Vancouver, BC, Canada*, Jun. 1998.

[3] M.Deering, "Geometry compression," *Proc. SIGGRAPH'95*, pp. 13–20, Aug. 1995.

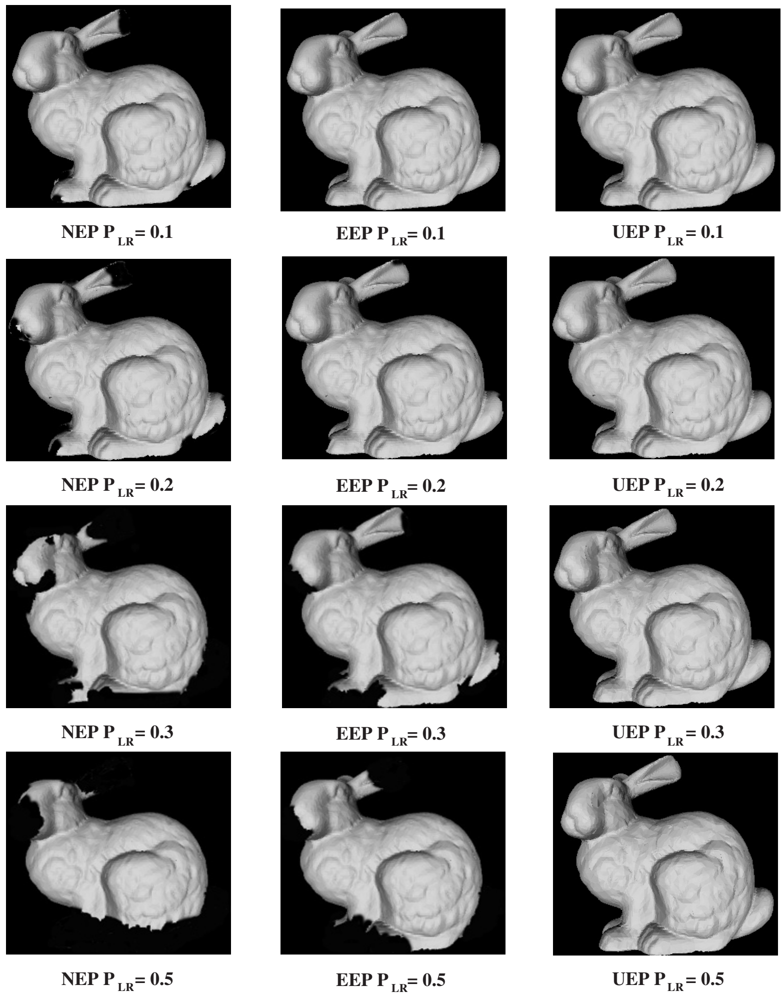


Fig. 8. Subjective results of applying no error protection (NEP), equal error protection (EEP), and unequal error protection (UEP) methods on SMALL mesh model. The caption under every image gives the error protection method and the channel packet loss rate. RS code (n, k) = (63,45)

[4] J. Rossignac, "Edgebreaker: connectivity compression for triangle meshes," *IEEE Trans. Vis. and comput. Graphics, Vol. 5, No. 1*, pp 47–61, Jan. 1999.

[5] M. Chow, "Optimized geometry compression for real-time rendering," *Proc. IEEE Vis'97, Phoenix, AZ*, pp 347–354, Oct. 1997.

[6] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal, vol. 29*, pp. 147–160, April 1950.

[7] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of Society for Indus. and Applied Maths*, Jun. 1960.

[8] P. C. Cosman, J. K. Rogers, P. G. Sherwood, and K. Zeger, "Combined forward error control and packetized zero tree wavelet encoding for transmission of images over time-varying channels," *IEEE Trans. Image Processing, 9(6)*, pp. 982–993, Jun. 2000.

[9] K. Sohn, J. R. C Lee, and W. Jang, "Error-resilient zerotree wavelet video coding," *SPIE Journal of Optical Engineering*, pp. 2480–2488, Nov 2001.

[10] G. Al-Regib, Y. Altunbasak, and J. Rossignac, "An unequal error protection method for progressively transmitted 3-d models," *IEEE Transactions on Multimedia, vol. 7, no. 4*, pp. 766–776, 2005.

[11] 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 Journal on Selected Areas in Communications, vol.18, no.6*, pp.819–828, 2000.

[12] S. Bischoff and L. Kobbelt, "Toward robust broadcasting of geometry data," *Comput. and Graph., vol.26, no.5*, pp.665–675, 2002.

[13] C.L. Bajaj, S. Cutchin, V. Pascucci and G. Zhuang, "Error Resilient Transmission of Compressed VRML," TICAM, The Univ. Texas at Austin, Austin, TX, Tech. Rep., 1998.

[14] Z. Yan, S. Kumar and C. Kuo, "Error resilient coding of 3-D graphic models via adaptive mesh segmentation," *IEEE Trans. Circ. Syst. Video Tech., vol.11, no.7*, pp.860–873, July 2001.

[15] G. Al-Regib and Y. Altunbasak, "An unequal error protection method for packet loss resilient 3D mesh transmission,," *IEEE INFOCOM*, 2002.

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

[17] M. Lounsbery, "Multiresolution analysis for surfaces of arbitrary topological type," *PhD. Thesis, University of Washington*, 1994.

[18] C. Christopoulos, A. Skodras, and T. Ebrahimi., "The jpeg2000 still image coding system: an overview." *IEEE Trans. Consumer Electronics Vol 46, Issue 4*, pp. 1103–1127, 2000.

[19] P. Schroder and W. Sweldens, "Spherical wavelets: Efficiently representing functions on the sphere," *in Proc. ACM SIGGRAPH*, pp. 161–172, 1995.

[20] W. J. Schroeder., "Decimation of triangle meshes." *in Proc. ACM SIGGRAPH*, pp. 65–70, 1992.

[21] C. Pimentel and I. Blake, "Modeling burst channels using partitioned fritchman's markov models," *IEEE Trans Veh Tech., vol. 47, no. 3*, pp.885–899, 1998.