IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 12, 2019, accepted December 8, 2019, date of publication December 16, 2019,
date of current version December 30, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2960053

A Three-Stage Progressive Transmission Scheme for Virtual Environments Over Lossy Networks

YITONG LIU, (Member, IEEE), JINGFENG GUO, KEN DENG, AND YISHI LIU
School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Yitong Liu (liuyitong@bupt.edu.cn)

ABSTRACT The rapid development of Virtual Environments (VEs) gives rise to its prominent performance in a myriad of multimedia applications. Accompanied by this is the increasingly massive data used for ascending user's visual experience. Recently, 3D streaming makes it viable to put VEs into real-time use through progressive transmission. Nonetheless, downloading VEs over inevitable lossy networks still remains a major bottleneck that urgently needs to be solved. In this paper, we propose a three-stage progressive transmission scheme for VEs over lossy networks, which aims to minimize distortion even when encountering poor network condition. The first stage is the Pre-coding Stage that offers the most feasible transmission scheme for subsequent transmission, in this stage we combine the advanced interest management algorithm and our proposed scene distortion estimation algorithm to bring up a scene update strategy which takes the user's visual features, quality of the scene and bandwidth estimation into account, so as to determine the download priority comprehensively. In the following Transmitting Stage, we apply Transmission Control Protocol (TCP) to make sure that we convey the essential information. More importantly, we adopt expanding window fountain (EWF) codes to provide an unequal error protection (UEP) for data with different importance. Eventually in the third After-process Stage, we employ a geometry prediction algorithm for the scenario with packet loss. The final results of our experiments show that the three-stage scheme can offer preeminent visual experience especially when it comes to a relatively inferior network condition.

INDEX TERMS Virtual environments, virtual reality, progressive transmission, quality of experience.

I. INTRODUCTION

The arrival of 5G provides significantly higher data bandwidth and low-latency communications, which largely accelerate the deployment of Virtual/Augmented Reality (VR/AR), Internet of Things (IoT) [1]–[3] Internet of Vehicles [4] and so on. On account of the incessant development of VR, Virtual environments (VEs) have been put into a myriad of practical uses [5], [6], from distributed simulation to online shopping, from virtual museums to online games. Although VEs are providing users with more and more stellar immersive experience, they are facing the challenge of dealing with great data size of 3D models. Currently, it is inevitable to download the data to the local in advance [7], while this whole process costs a long period and prodigious storage space. With the dramatic increase of scale and complexity of VEs, the contradictions between real-time requirements and limited bandwidth, huge data size and limited storage space have become the key baffle restricting the further development of VEs.

Currently, many studies transmit VEs progressively using 3D streaming [8]–[10], [12], [14], which is similar to video streaming [13]. Reference [8] presents a method of 3D streaming for collaborative design in networked environment which improves the efficiency, effectiveness and security in sharing large CAD files over the network. Reference [10] propose an original progressive representation of branching systems adapted to the streaming of 3D scenes. Reference [12] use DASH on networked VEs for 3D content streaming, which make intelligent decisions about how to download, balancing between geometry and texture while being adaptive to network bandwidth. 3D streaming aims to incessantly

The associate editor coordinating the review of this manuscript and approving it for publication was Dapeng Wu.

transmit 3D contents over a network for the real-time purpose. Instead of downloading the entire scene at the cost of time efficiency, user only needs to download the data used for rendering the given scene before navigating. The salient aspect of progressive transmission is that it reduces the data size requested each time, which benefit the usage of storage space [15]. The progressive transmission of VEs is mainly reflected in two aspects: (1) Progressive transmission of the scene. Because of user's visual limitation, only the scene user interested in is gradually obtained. (2) Progressive transmission of the 3D model. Through multi-resolution coding, the 3D model with different resolutions will be downloaded on demand. To a better transmission of VEs, some studies concentrate on P2P (peer-to-peer) network [8]–[11], [16]. Hu *et al.* [16] propose a framework for 3D streaming on P2P networks called FLoD, which allows clients of VEs applications to obtain relevant data from other clients while minimizing server resource usage.

As mentioned above, we can achieve real-time VEs use through progressive transmission methods over limited bandwidth. However, downloading VEs over lossy networks still remains a major bottleneck. Lossy network can actually be a severe while inevitable problem during transmission process. For wireless network, wireless medium access is consistently affected by background noise, multi-path fading, shadowing and interference. Causing bandwidth to fluctuates drastically over time, which leads to link disruptions and thereby results in high error rates and packet loss [19]–[21]. Technically, packet loss is unavoidable, retransmission of all the lost packets will cause unbearable delay. So, fast delivery of VEs over lossy networks is still a crucial task due to limited bandwidth and packet loss.

We have previously proposed a transmission framework for 3D virtual scenes in a bandwidth-limited network [22]. Based on the previous results, this paper considers the situation of network packet loss and combines the data characteristics of the 3D mesh model. Peer-to-peer error protection, and a geometric data prediction algorithm is proposed to make the proposed framework better adapt to the lossy network.

In this paper, we propose a three-stage VEs progressive transmission scheme. In the Pre-coding Stage, we firstly bring forward an advanced interest management algorithm, which enables AOI to better adapt to user's tiny movement and personal behavior, thereby utilizing the bandwidth in a more efficient way. Then we bring up a particular scene distortion estimation algorithm on the basis of user's visual features. These two algorithms contribute to our proposed update strategy, which determines the priority of models' transmission. Currently, scholars usually decide the transmission priority through the information such as visual importance, reusability, times of attention and so on [24]. While our update strategy considers the network bandwidth fluctuations and scene distortion. In the Transmitting Stage, we apply Transmission Control Protocol (TCP) to make sure that we convey the essential information. More importantly, we adopt expanding window fountain (EWF) codes to provide an unequal error protection (UEP) for data with different importance. In the After-process Stage, a geometry algorithm is applied to cope with packet loss. Through the download of relevant 3D models with a proper resolution, we can finally achieve a salient utilization of limited bandwidth, therefore minimizing distortion. The architecture of the proposed scheme is depicted in Fig.1. The experimental results show that the proposed scheme can lower distortion especially when it comes to a relatively inferior network condition.

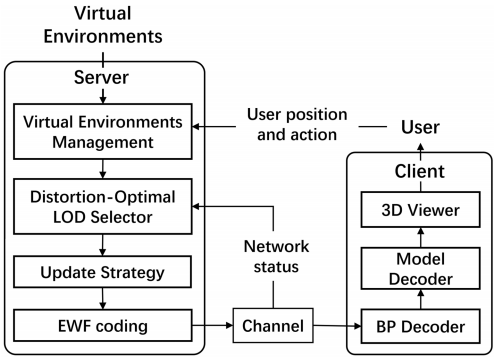

FIGURE 1. Architecture of the proposed VEs progressive transmission scheme.

The reminder of this paper is organized as follows. Section II presents related works. In section III, we explicitly introduce our Pre-coding Stage cooperating with Virtual Environments management and distortion-optimal LoD selector. We analyze Transmitting Stage and After-process Stage in section IV and section V respectively. The section VI gives the conclusion.

II. RELATED WORKS

In this section, we anaylze some related works, which include 3D streaming, interest management, distortion-awareness algorithm and EWF codes.

A. 3D STREAMING

3D streaming achieves a continuous, real-time transmission of 3D contents, such as meshes, textures, animations and scene graphs, allowing users to interact without downloading the whole scene. Currently, 3D streaming can be divided into four types: object streaming, scene streaming, visualization streaming and image-based streaming [9]. In this paper, we mainly focus on scene streaming, which is extended from object streaming. Now, we will analyze object streaming and scene streaming in details.

In object streaming, 3D model is pre-coded and divided into a base layer and a series of refinement layers. After downloading the base layer, user can view the model. With the downloading of refinement layers, the model is gradually refined. Progressive meshes (PM) is one of the most practicable object streaming [25], [26] algorithms. PM aims to simplify the 3D model through reducing the number of vertices and faces after a series of edge collapse operations.

The 3D model is stored as a base mesh and a series of refinement meshes to achieve progressive loading of the model. After a series of vertex-split operations, the model can be reconstructed appropriately. These two processes are shown in Fig.2.

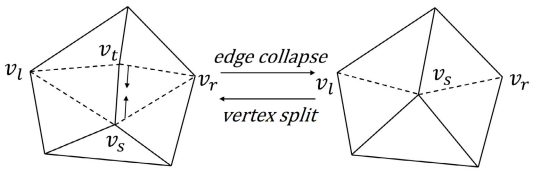

FIGURE 2. Edge-collapse and vertex-split operations.

Diverse scene streaming algorithms have been brought up to transmit the data of VEs, including client-server [27], [31] and peer-to-peer [8]–[10], [16]–[18] architectures. Although these two architectures download the data source distinctively, they both transmit the 3D contents to meet the user's need by calculating the user's interest and visual importance of the objects. Scene streaming is an extension of object streaming which extends a single 3D model to an entire scene. The 3D models are placed at any position within the VEs. Due to the occlusion between 3D models and visual limitation, only partial scene need to be downloaded.

B. INTEREST MANAGEMENT

In scene streaming, interest management determines the user's visible objects. [28]–[30]. Interest management is implemented by identifying the AOI. The AOI can be a circular area (Fig.3 (1)) with the user's viewpoint as the center and visible distance as the radius. Objects within the AOI are considered as the current set of visual scenes and have the highest priority for downloads. When entering a new scene, users only need to obtain the scene collection within the AOI without downloading the entire scene collection. This not only solves the problems of the requirements for available bandwidth and storage space in real-time viewing, but also greatly reduces the need for real-time computing visibility. Several common AOIs are compared in detail in [28], which proposes the A3 AOI, as shown in Fig.3 (2). The A3 AOI segments the AOI into field of view and close area, leading to an efficient use of bandwidth.

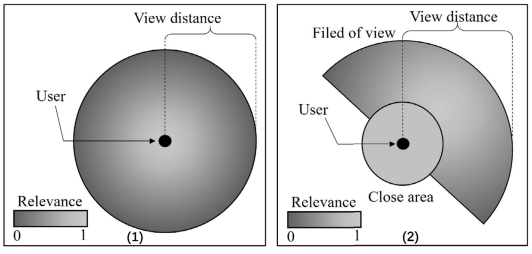

FIGURE 3. Circular area of interest and A3 area of interest.

C. DISTORTION-AWARENESS ALGORITHM

Currently, estimating the distance between discrete 3D surfaces represented by triangular 3D meshes mainly uses Hausdorff distance. The main reason for using the Hausdorff distance to measure the difference between two meshes is that MSE and PSNR cannot measure the difference between meshes with different connectivities. The Hausdorff distance measures the largest minimum distance between points on two surfaces, so there is no need for a correspondence between the vertices of the two meshes. In a virtual scene, the model of different locations affects the user differently, In Cyberwalk [18], Chim et al. proposed the concept of visual importance for the optimal resolution, where an object closer to the viewer has higher visual importance.

D. EWF CODES

Expanding window fountain (EWF) codes [23] is one of the UEP schemes based on Luby Transform (LT) code [32]. EWF encoding proceeds in a slightly different fashion than the usual LT encoding. As shown in Fig.4, the input k symbols are divided into n levels according to their importance, and the symbol amount of *i-th* level is s_i, we denote the size of the *i-th* window as k_i, where $k_1 < k_2 < \ldots < k_n = k$. Each level defines a window and is included in the next window. The most important symbols are set in the innermost window, meanwhile, the outermost window contains all the k symbols. Upon the window selection, a new encoded symbol is determined with an LT code described by the selected window degree distribution as if encoding were performed only on the input symbols from the selected window. Obviously, EWF code design generalizes the standard LT code design as LT codes are EWF codes defined by a single window, i.e., all the input symbols are of equal importance.

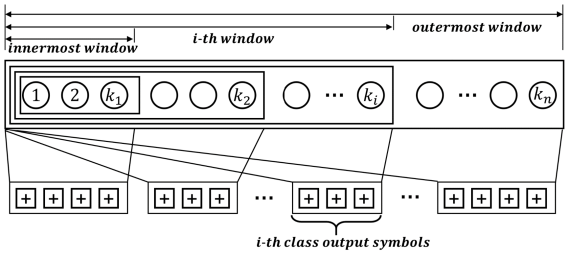

FIGURE 4. Expanding window fountain (EWF) codes.

III. PRE-CODING STAGE

Based on the analysis and summary of the current VEs transmission strategy, this paper proposes a three-stage progressive transmission scheme for VEs over lossy networks. The detailed transmission scheme framework is shown in Fig.5.

First and foremost, we need to make sure what is going to be transmitted in this process to better fit user's need and ascend the whole visual experience, which means to minimize distortion. Thus, we bring up Pre-coding Stage consists of three segments called virtual environments management, distortion-optimal LoD selector and update strategy.

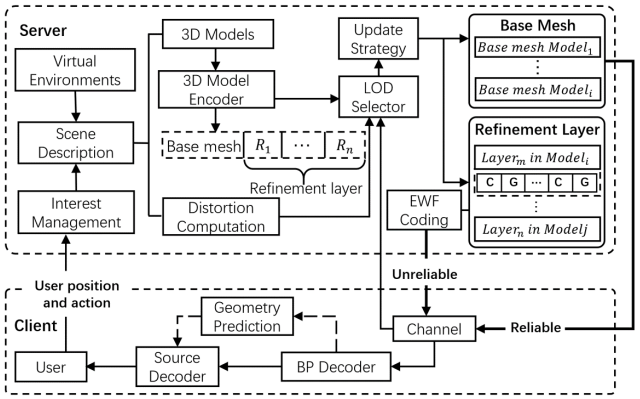

FIGURE 5. Architecture of the proposed VEs progressive transmission scheme in detail.

A. VIRTUAL ENVIRONMENTS MANAGEMENT

In this part, the basic information of VEs is described as a scene description. Each 3D model may be progressively encoded into a base layer and a series of refinement layers, by which the user can request for relevant models with different resolutions. When viewing the scene, the user can determine the relevant models through scene description and interest management.

B. DISTORTION-OPTIMAL LOD SELECTOR

The users can observe an object that is closer to the user with an higher resolution. Based on this theory, the scene distortion computation algorithm is proposed. Associated with the bandwidth estimation, the optimal LOD (Level of Detail) of each relevant model is aimed for minimizing the scene distortion while present reconstructed models of higher quality.

1) UPDATE STRATEGY

After the optimal level of detail of the relevant model is determined by the Distortion-Optimal LOD selector, and the importance level of each optimization layer of the corresponding model is determined, the download priority of each optimization layer is determined, thereby completing the progressive update of the scene.

C. VIRTUAL ENVIRONMENTS MANAGEMENT

Virtual Environments Management can be accomplished through three steps: Scene Description, 3D Model Encoder and Interest Management.

1) SCENE DESCRIPTION

We abstracts the VEs into a 2D scene description, as shown in Fig.6. During viewing the scene, the scene falling into the user's AOI is the relevant scene. The computational complexity of this process can be greatly reduced if combined with scene description. In order to evaluate the relevant 3D models more efficiently, we divide the VEs into several sub-area blocks with same size, where each sub-area has a fixed id.

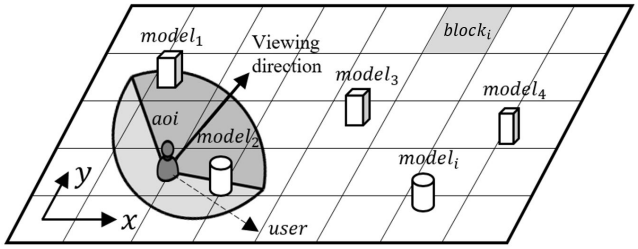

FIGURE 6. Scene description of virtual environments.

3D models are divided into different blocks according to their respective locations.

The scene description contains the following information: (shown in Fig.7) the models' *id*, horizontal projection position ($location_x$, $location_y$), block's *id* which indicating the model's location, size of the 3D model's data, the information of the base layer and the refinement layers.

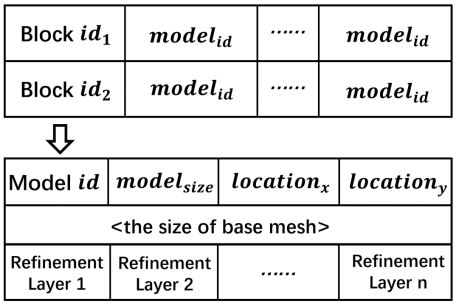

FIGURE 7. The information contained in scene description for virtual environments.

2) 3D MODEL ENCODER

Progressive meshes (PM) [25] intends to simplify and encode 3D models, as depicted in Fig.8. We encode the 3D model to a base mesh and a series of refinement layers, which consist of connectivity and geometric data, in a progressive manner. The decoder can rebuild the model at arbitrary resolution through vertex split, which is the reverse operation of edge collapse, when it comes to a given base mesh. Each vertex split process record is a 6-byte quantity [25], as shown in Table 1.

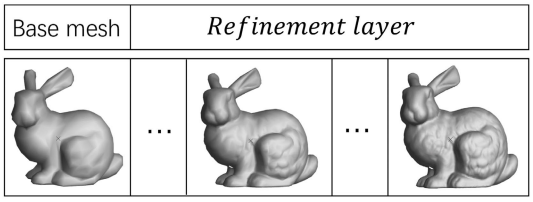

FIGURE 8. The structure of a progress mesh.

Assuming *size* and $size_t$ represents the initial 3D model's data size and the reconstructed model's data size respectively,

TABLE 1. Vertex split process.

Connectivity data (bits)			Geometric data (bits)
$\Delta flclw$	vs_index	vlr_rot	$vt\_coordinates$
7	2	3	36

then the *lod* can be described by

$$lod = \frac{size_t}{size} \quad (0 \le size_t \le size, 0 \le lod \le 1) \tag{1}$$

3) INTEREST MANAGEMENT

The proposed AOI is shown in Fig.9. The AOI is distributed into field of view and relevant area. Field of view stands for user's current visual field, and the relevant area can avoid the situation of sudden blank. We prescribe the visual angle to be $\frac{2}{3}\pi$, the visible radius to be $view_r$, the angle that line of sight deviates from the x-axis to be θ, $\theta =< \vec{x}, \vec{v} >$, the radius of the relevant area to be $relevant\_area_r$, whose relationship with $view_r$ can be expressed as:

$$relevant\_area_r = view_r \cdot e^{-|cos\frac{3\gamma}{4}|} \quad (\frac{2}{3}\pi \le \gamma \le 2\pi) \tag{2}$$

where γ represents the angle between the close area and the lower bound of the field of view, $\gamma =< \vec{b}, \vec{m} >$.

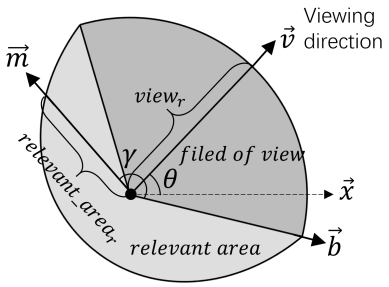

FIGURE 9. Area of interest-filed of view and relevant area.

We define the radius of the relevant area considering the following three items:

1) Diminishing the quantity of related models in comparison with the circular area (Fig.3(1)) to decrease the quantity of data per request. Therefore, the models in the field of view are provided with better definitions.

2) The relevant area's radius fluctuates as the time goes. It can prevent the user from the situation of sudden blank.

3) The user's head rotation appears apparently to be a continuous process, so that users tend to rotate a smaller angle. On account of that, the radius of the relevant area is smaller when deviation angle is larger.

D. DISTORTION-OPTIMAL LOD SELECTOR

The user's AOI includes multiple models. The requirements for each model vary due to the variance in distance, and angel of different models with respect to the user. We do not need to transfer all the layers for each model. In addition, owing to the fact that networks are constantly fluctuating, balancing models' LOD to provide the optimal overall viewing experience can be really crucial. We put forward a scene distortion estimation algorithm on the basis of user's visual characteristics to evaluate the distortion. With the minimum distortion and a more reasonable allocation of bandwidth as the optimization targets, the LOD selector is proposed.

1) DISTORTION COMPUTATION

The model with different LOD can be reconstructed based on the base mesh, repectively. The error between the model with various LOD and the initial one is determined as the distortion for this model. Hausdorff distance is actually a practical way to gauge 3D models' distortion. Accordingly, the distortion between the model with lod_i and lod_j in Hausdorff distance can be written as:

$$D_{i,j} = \max\{h(m\_lod_i, m\_lod_j), h(m\_lod_j, m\_lod_i)\} \tag{3}$$

$$h(m\_lod_i, m\_lod_j) = \max_{e \in m\_lod_i} \min_{f \in m\_lod_j} ||e - f|| \tag{4}$$

where $||\cdot||$ means the Euclidean distance, m_lod_i means the model with LOD i. Hence, the distortion gauges the maximum distance between these two models.

Through calculating the distortion value between the model with different LOD and the original one, we can obtain a distortion curve. The functional relationship between the distortion and Level of Detail is described with the curve. The model distortion function can be represented as follows:

$$D_{lod,0} = \frac{\mathrm{a}}{\mathrm{b} + lod} \quad (0 \le lod \le 1) \tag{5}$$

where a and b are constants calculated during the encoding of the models, and lod represents the LOD of the rebuilt model. a and b are calculated when models are encoded. Fig.10 shows the Lod-Distortion curves of the classic models

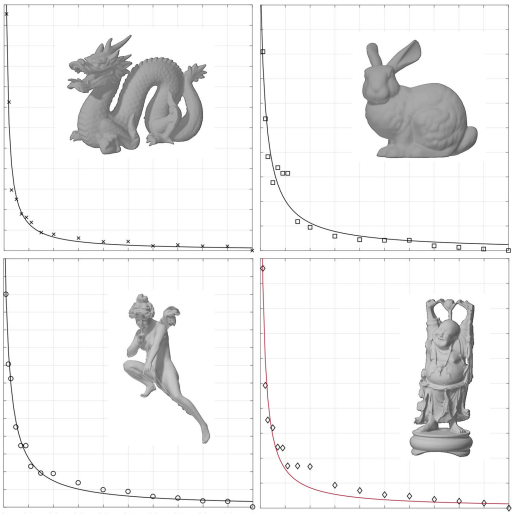

FIGURE 10. The distortion of 3D model and the rebuilt models with various resolution.

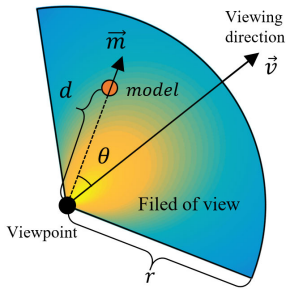

FIGURE 11. The parameters in the definition of the viewing importance.

(Dragon, Stanford Bunny, Angel and Happy Buddha). As can be seen, the Equation (5) fits the single model distortion well.

User can have a clearer view of the model (i.e. with higher resolution) with smaller deviation angle and closer distance. Even the same model with same LOD may have different observation results by the user, if in diffrenent positions. So we define the visual importance according user's visual characteristics in Equation (6).

$$I=\beta(1-\frac{d}{r})+(1-\beta)cos\frac{\theta}{2} \quad (0\le d\le r, 0\le\theta\le\frac{\pi}{3}) \quad (6)$$

where β stands for a constant, $\beta \in (0, 1)$, d stands for the Euclidean distance between the model and the viewpoint, r stands for the radius of field of view, θ stands for the angle between the model and the user's viewing direction, $\theta =< \vec{v}, \vec{m} >, \theta \in [0, \pi/3]$.

Fig.12 gives the distribution of visual importance. The color indicates the visual importance, the lighter the color, the higher the visual importance. It is obviously depicted that the significance of models in field of view is way higher than models in relevant area. Models that exist in the same area own visual significance determined by two major elements, one is the geometric distance, and the other is the angle of deviation from the user's viewing direction. So we can safely arrive at the conclusion that when the model is closer to the user and the angle from the user's viewing direction is smaller, the model is more essential.

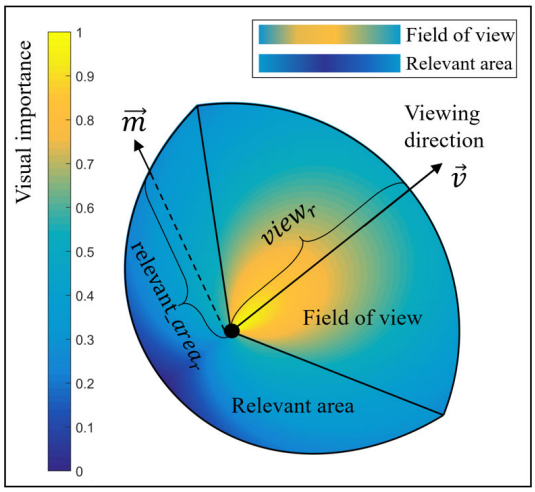

FIGURE 12. Distribution of visual importance in area of interest.

Taking the influence of distance and angle to the visual importance into account, we can obtain the scene distortion with the following equation:

$$Distortion=\sum_{i=1}^{N}[\beta(1-\frac{d_i}{r})+(1-\beta)cos\frac{\theta_i}{2}]D_{i_{lod,0}} \quad (7)$$

where N stands for the quantity of models in field of view, $\beta(1-\frac{d_i}{r})+(1-\beta)cos\frac{\theta_i}{2}$ stands for the *i-th* model's visual importance, $D_{i_{lod,0}}$ stands for the distortion of the model where the LOD is lod.

2) LOD SELECTOR

In the previous section, the scene distortion function has been defined. As stated in the formula, the range of effects of distortion is determined by the 3D models in the user's field of view. No direct relationship exists between distortion and the 3D models in other regions. We can estimate the download budget by bandwidth estimation, which has been studied a lot before. Each relevant model's resolution has a corresponding influence on the scene distortion. We aim to determine the optimal LOD of each model that improve user's experience without adding any extra burden to the download budget. The problem posed above can be expressed by the following formula:

$$\begin{cases} \min\{f(lod)=\sum_{i=1}^{N}[\beta(1-\frac{d_i}{r})+(1-\beta)cos\frac{\theta_i}{2}]\frac{\alpha_i}{lod_i+b_i}\} \\ \sum_{i=1}^{N} lod_i\cdot data_i \le data_{es} \\ 0\le lod_i\le 1, \quad i=1,2,3,\ldots,N \end{cases} \quad (8)$$

where $data_{es}$ is the download budget, $f(lod)$ represents the distortion function of the current scene in the user's field of view, N represents the quantity of 3D models in field of view, and $data_i$ represents the amount of data of the *i-th* 3D model.

We adopt the Lagrangian multiplier method to solve this function and construct the following function:

$$\begin{cases} \min\{f(lod)\} \\ h(lod)=\sum_{i=1}^{N} lod_i\cdot data_i - data_{es} \\ L(lod,\lambda)=f(lod)+\lambda\cdot h(lod) \end{cases} \quad (9)$$

After solving the equations, we can get the result as:

$$\begin{cases} \frac{\partial L}{\partial lod_1}=-\frac{\mu_1\cdot a_1}{(lod_1+b_1)^2}+\lambda\cdot data_1=0 \\ \frac{\partial L}{\partial lod_2}=-\frac{\mu_2\cdot a_2}{(lod_2+b_2)^2}+\lambda\cdot data_2=0 \\ \quad\vdots \\ \frac{\partial L}{\partial lod_N}=-\frac{\mu_N\cdot a_N}{(lod_N+b_N)^2}+\lambda\cdot data_N=0 \\ \frac{\partial L}{\partial\lambda}=\sum_{i=1}^{N} lod_i\cdot data_i - data_{es}=0 \end{cases} \quad (10)$$

where

$$\mu_i = \beta(1 - \frac{d_i}{r}) + (1 - \beta)cos\frac{\theta_i}{2} \tag{11}$$

According to the Equation (10), the optimal LOD of each relevant 3D models that minimize the scene distortion without augmenting the download budget can be calculated.

When the user moves, there will be a partially cached model locally, and the function will be:

$$\begin{cases} \min\{f(lod) = \sum_{i=1}^{N}[\beta(1-\frac{d_i}{r}) + (1-\beta)cos\frac{\theta_i}{2}]\frac{\alpha_i}{lod_i + b_i}\} \\ \sum_{i=1}^{N}(lod_i - lod_{i,0}) \cdot data_i \le data_{es} \\ lod_{i,0} \le lod_i \le 1, \quad i = 1, 2, 3, \ldots, N \end{cases} \tag{12}$$

The solution process is the same as when there is no cache.

E. UPDATE STRATEGY

Considering that the correlation between sub-layers, the transmission reliability needs to be improved. Especially in lossy networks, the packet loss will have a great impact on the quality of the transmission. After progressive meshes encoding, there is a strong dependency between data. Once a part of data is lost in transmission, the data that depends on this part cannot be decoded even if it is correctly received. Considering the real-time requirements, data that cannot be decoded normally will be discarded, which will result in waste of bandwidth resources. So in order to achieve a balance between real-time and reliability, we combine hybrid protocols (3TP) [33] and UEP based on characteristics of the 3D model data. In "3TP", data that more significant transmitted using Transmission Control Protocol (TCP), while the less important data transmitted through User Datagram Protocol (UDP) to decrease latency.

In our scheme, the transmission method similar to the TCP/UDP hybrid protocol is adopted. The initial 3D model is stocked as a series of sub-layers through PM, and the index of the base layer is set to 0. Refinement layers' index is increased sequentially. It can be seen from Fig.10 that, the model's low-level data has stronger influence on the model's distortion, therefore we segment the remaining data into refinement layers unequally. Typically, the data of base mesh model is small (usually less than 5% of the model), while the refinement layers occupy the majority data size. However, The base layer is the basis for all subsequent refinement layers, so the base mesh is the most important portion of the 3D model and it should be transmitted using reliable transmission (TCP). In this way, the client can receive the base mesh data accurately and completely. The user can view the outline of the model once the base layer is decoded and rendered. The refinement layers are transmitted over unreliable channel (UDP) to reduce the delay.

It can be seen from Fig.13 that, after receiving the sublayer j of *i-th* model, the decoded mesh's distortion would decrease from the sublayer ($lod_{i,j-1}$)'s distortion to the next sublayer ($lod_{i,j}$)'s distortion. We get the distortion's diminution amount and increase LOD of sublayers in the refinement through Equation (13) and Equation (14) respectively.

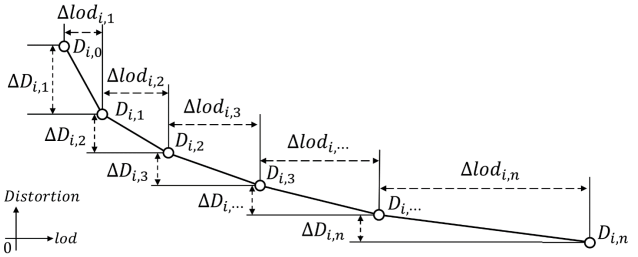

FIGURE 13. The distortion's diminution amount of the sublayer ($lod_{i,j-1}$) to the distortion of the next sublayer ($lod_{i,j}$).

$$\Delta D_{i,j} = D_{i,j} - D_{i,j-1} \quad (j \ge 1) \tag{13}$$

$$\Delta lod_{i,j} = lod_{i,j} - lod_{i,j-1} \quad (j \ge 1) \tag{14}$$

It appears that we ought to firstly transmit the sub-layer that owns a higher distortion diminution. But when the size of *j-th* sub-layer of *i-th* model, $size_{i,j}$, shown in Equation (15), is relatively huge, we may spend a long period on completing the transmission process. Taking distortion diminution and data size into account, we obtain the definition of the relative importance for *j-th* sub-layer $R_{i,j}$ in Equation (16). $R_{i,j}$ stands for distortion diminution amount per unit data size for sub-layer j of *i-th* model. Actually, larger $R_{i,j}$ may cause higher distortion diminution per unit data size and therefore ought to be firstly transmitted.

$$size_{i,j} = \Delta lod_{i,j} \cdot size_i \tag{15}$$

$$R_{i,j} = \Delta D_{i,j}/size_{i,j} \tag{16}$$

The LOD of each relevant 3D model can be defined with the assistance of LOD selector. Also, each relevant model's index of sub-layers can be obtained through the calculated LOD. Based on each sub-layer's distortion diminution per unit data size, we can get the 3D model distortion reduction matrix as follow:

$$\mathbf{R} = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,level_1} & \cdots \\ R_{2,1} & R_{2,2} & \cdots & R_{2,level_2} & \cdots \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ R_{N,1} & R_{N,2} & \cdots & R_{N,level_N} & \cdots \end{bmatrix} \tag{17}$$

where $level_i$ represents the optimal number of layers of the *i-th* 3D model.

$R_{i,j}$ in matrix (17) indicates the distortion diminution per unit data size for sub-layer j of *i-th* model. The model's sub-layer with the largest $R_{i,j}$ in matrix is preferentially requested. In the meanwhile, sub-layers from the same model will be requested in order. The field of view priority principle should be followed when updating the scene.

IV. TRANSMITTING STAGE

After determining the download priority by the update strategy, we use EWF codes to provide an unequal error protection (UEP) for models' data with different importance.

As shown in Table 1, the refinement layers' data contains two parts, connectivity data and geometric data. The connectivity data records the connection between the new vertex and the current edge. The loss of this part information will directly lead to the termination of the model decoding, and the model will stay at the current resolution. The geometric data records the vertex's coordinate. If lost, we can estimate it based on existing information, and a small deviation is acceptable. So the refinement layers are transmitted over unreliable channel (UDP) to reduce the delay.

The connectivity and geometric data are separated and protected by EWF code. The amount of geometric data is about three times that of the connectivity data. As shown in Fig.14, the input symbols is divided into two parts according to the importance, MIB (More Important Bits) and LIB (Less Important Bits), corresponding to connectivity and geometric data. Considering the data proportion in this scenario, assume that there are a total of k input symbols, and the number of MIB is k_1, k_1:$k = 1:4$, the number of LIB is k-k_1. We define two windows at the same time, the first window is w_1, corresponding to MIB, the second window is w_2, corresponding to the total data. The number of output symbols is n, and the overhead is defined as $\varepsilon = n/k - 1$.

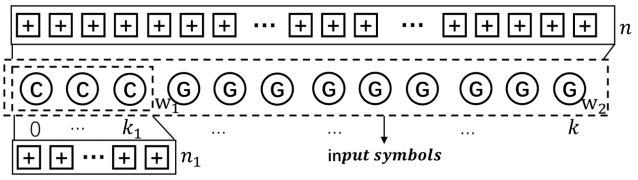

FIGURE 14. The unequal error protection method for 3D meshes based on EWF. "C" represents connectivity data and "G" represents geometric data.

In LT encoding process, the encoder selects several information symbols for XOR operation according to a certain probability. Usually, the number of information symbols involved is called the degree of code symbol. In this paper, we use RSD (Robust Soliton Distribution):

$$\Omega(i) = \frac{\rho(i) + \tau(i)}{\beta} \tag{18}$$

where:

$$\rho(i) = \begin{cases} 1/k & i = 1 \\ 1/[i(i-1)] & i = 2, 3, \ldots, k \end{cases} \tag{19}$$

$$\tau(i) = \begin{cases} S/(k \cdot i) & i = 1, 2, \ldots, (k/S) - 1 \\ S/k \cdot ln(S/\sigma) & i = (k/S) \\ 0 & i = k/S + 1, \ldots, k \end{cases} \tag{20}$$

$$\beta = \sum_{i=1}^{k} (\rho(i) + \tau(i)), \quad S = cln(k/\sigma)\sqrt{k} \tag{21}$$

k is the number of input symbols, σ is the largest decoding failure probability, and c is a constant not greater than 1. The RSD is shown in Fig.15. when $k = 10000$, $c = 0.2$, $\sigma = 0.05$.

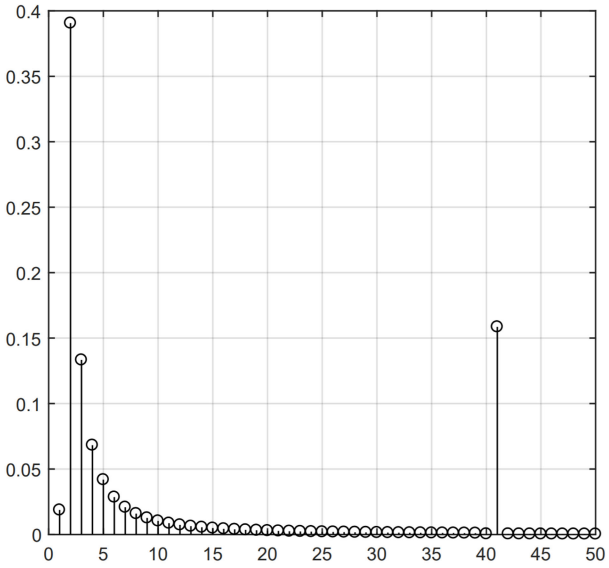

FIGURE 15. Robust soliton distribution.

The coding rules of the EWF for 3D models are illustrated as follows:

1) Divide the k input symbols into MIB and LIB corresponding to connectivity and geometric data. The amount of MIB is k_1, and LIB is k-k_1.
2) Put the MIB symbols into w_1 window, MIB and LIB symbols into w_2 window.
3) Select the window w_1 with the probability Γ_1, and select the window w_2 with the probability (1-Γ_1).
4) Select a degree d according to the degree distribution. In selected window, chose d symbols randomly and uniformly *XOR* to get the encoded symbols.
5) Repeat the process 3-4 until the encoding is done.

Considering that the MIB accounts for 0.25 of the total data, taking $\Gamma_1 = 0.2$, $\Gamma_2 = 0.8$. Fig.16 shows Bit Error Rate (BER) of MIB and LIB under different overheads. It is significant to note that MIB symbols can be decoded much better than LIB. Under the same decoding overhead ε, the BER of LIB is higher than MIB. As the decoding overhead increases, the error performance of both MIB and LIB continues to increase. We note that as the overhead ε grows, the BER of both MIB and LIB decrease to 0.

Fig.17, Fig.18 and Fig.19 give the BER results of MIB, LIB and EEP (Equal Error Protection) at different Packet Loss Rate (PLR), taking $\Gamma_1 = 0.2$, $\Gamma_2 = 0.8$. The mean loss rate form 0 to 30%. The overhead ε of Fig.17 is 0.1, Fig.18 is 0.2, Fig.19 is 0.3. The pictures show that the BER of MIB is much lower than LIB and EEP. Therefore, the EWF coding can well guarantee unequal error protection performance for different important levels.

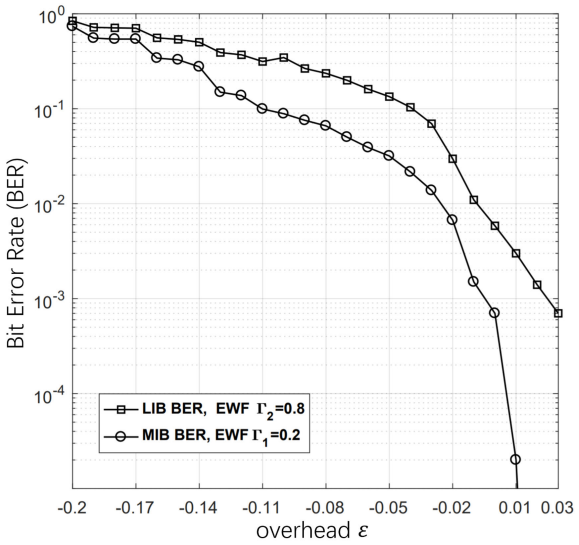

FIGURE 16. Asymptotic analysis of BER for MIB and LIB versus the overhead ε.

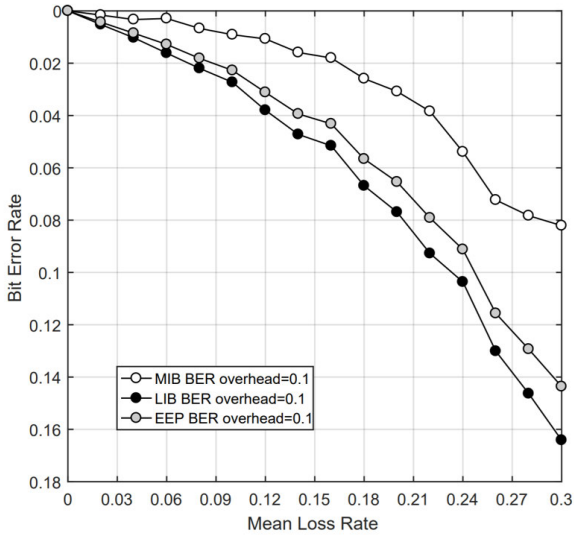

FIGURE 17. Asymptotic analysis of BER for MIB and LIB versus the PLR.

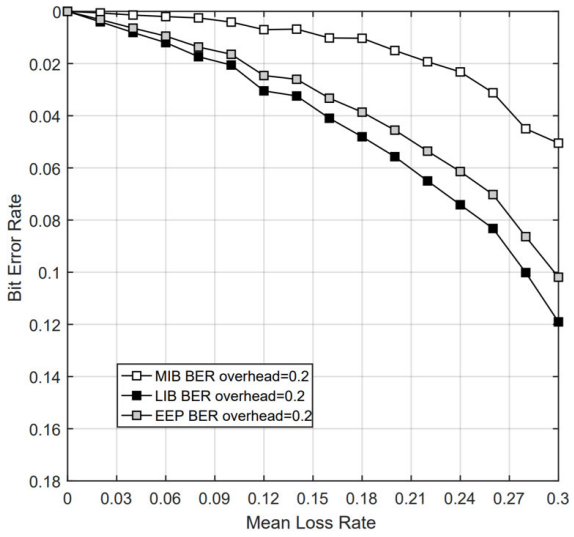

FIGURE 18. Asymptotic analysis of BER for MIB and LIB versus the PLR.

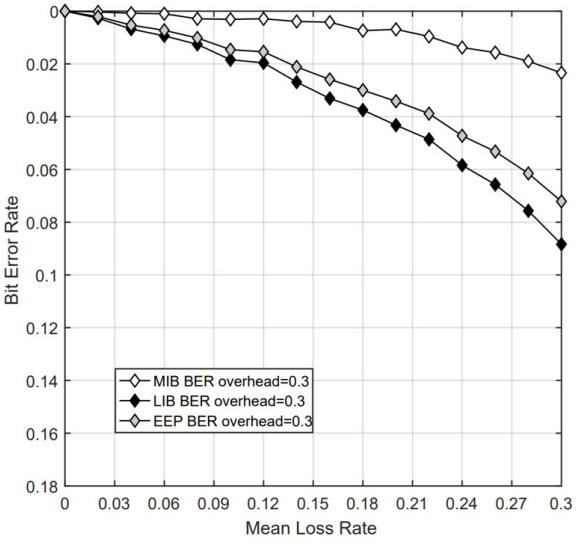

FIGURE 19. Asymptotic analysis of BER for MIB and LIB versus the PLR.

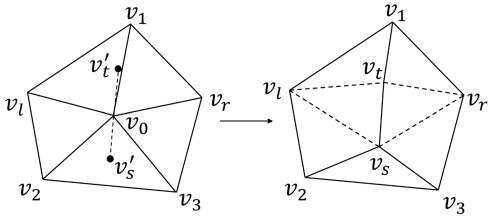

FIGURE 20. Geometry prediction process.

V. AFTER-PROCESS STAGE

For losing geometric data due to packet loss, a geometry prediction algorithm is employed to predict the coordinates of the new vertex.

In comparison,the use of EWF leads to a higher probability of losing geometric data, and it will affect the reconstruction of the 3D model. So a geometry prediction algorithm is employed to predict the coordinates of the new vertex. Due to the vertex split process is performed in a very small area, we can predict its coordinates with its neighbors using connectivity data. The algorithm is shown in Fig.20, v_0 is the split vertex, v_l is the left vertex of the collapse edge, and the v_r is the right vertex. We divide this area grid into the upper part and the lower part according to v_0, v_l, v_r. The prediction process is illustrated as follows:

1) If the number of triangles in one part is even like the upper part in Fig.16, select three vertices $\{v_l, v_1, v_r\}$ of the middle two triangles $\{\Delta v_l v_1\ v_0, \Delta v_0\ v_1\ v_r\}$.
2) If the number of triangles in one part is odd like the lower part in Fig.16, select three vertices $\{v_0, v_2, v_3\}$ of the middle triangle $\Delta v_2\ v_0\ v_3$.

3) Calculate the midpoint $v_t^{'}$ and $v_s^{'}$ of the three selected points $\{v_l, v_1, v_r\}$ $\{v_0, v_2, v_3\}$.
4) Calculate the midpoint v_t and v_s of the split vertex v_0 and the midpoint calculated in last process $v_t^{'}$ and $v_s^{'}$.

The result of Stanford Bunny model which consists of 34835 vertices and 69472 triangles using geometry prediction is presented. After progressive meshes encoding, the base mesh is about 5% of total data. We set the error rate of refinement layers 2%, it means that the coordinates of nearly 667 vertices is lost. The reconstructed model is shown in Fig.21(2), Fig.21(1) is the original model. It can be seen that there is only a slight difference. Using Hausdorff distance to calculate the distortion, it only increase about 0.1.

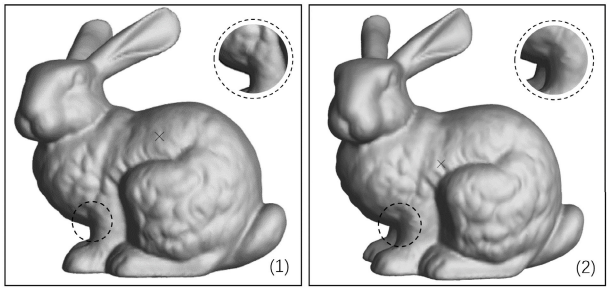

FIGURE 21. The original Bunny model and reconstructed model using geometry prediction.

VI. SIMULATION RESULTS

We also provide experimental results from our three-stage scheme. Table 2 gives the simulation parameters in detail.

A 1000 × 1000 VEs is simulated and separated into 400 sub-areas with a fixed size of 50 × 50. We spread these100 3D models arbitrarily over the entire scene, then encode them through PM. The base layer accounts for 5% of the total model data, and the remaining data is segmented into 12 refinement layers unevenly as shown in Table 2. We set the radius of field of view and the viewing angle to be 150 units and 120° respectively.

TABLE 2. Simulation parameters.

Parameters	Value
VE length (units)	1000
VE width (units)	1000
Block size (units)	50*50
Model size (MB)	0.5-1.5
Model number	100
Base mesh accounts for the whole model (%)	5
Refinement layer account for the whole model (%)	1,1,2,3,4,6,8, 10,12,14,16,18
View Distance (units)	150
View angle	120°

We can obtain the optimal LOD of models through LOD selector, the optimal LOD of each model under variable network status is shown in Fig.22. For the purpose of ignoring the influence of the data size, we adopt $Download\_Budget/Size_{sum}$ to represent the network status, the $Size_{sum}$ represents the total data size of models. It is apparently shown from the curves that there are four 3D models in user's field of view. When it comes to a relatively inferior network condition ($Download\_Budget/Size_{sum} \in (0, 0.2)$), the optimal LOD of each model fluctuates greatly. While when it comes to a relatively superior network condition ($Download\_Budget/Size_{sum} > 0.2$), the change becomes stable.

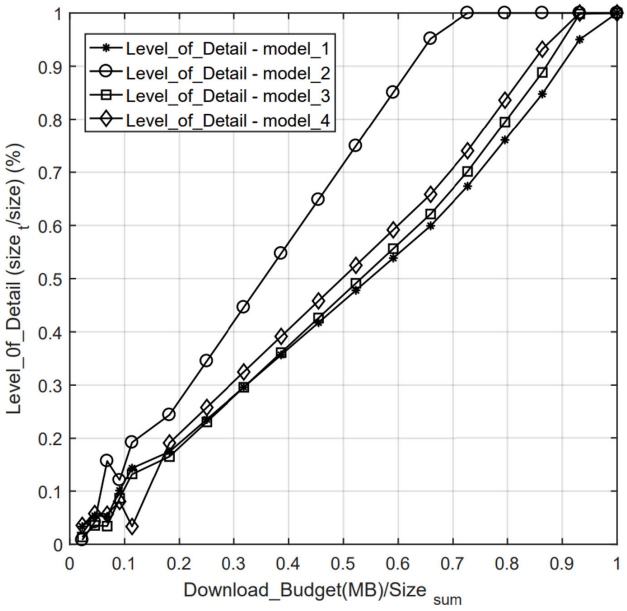

FIGURE 22. The optimal *Levle_of_Detail* of models in user's field of view with the change of $Download\_Budget/Size_{sum}$.

The update strategy is compared with the visual importance priority method when packet loss is zero. We contrast the distortion in users' field of view under the same network bandwidth. Fig.23 reveals the corresponding distortion results and distortion diminution which represents the divergence between these two methods. It is depicted by the curves that our three-stage scheme owns a distortion lower than the compared one, especially when $Download\_Budget/Size_{sum} \leq 0.25$, which is due to our reasonable allocation of bandwidth. When the $Download\_Budget/Size_{sum}$ rises, these two method perform closer on account that the model's high-level data effects the model distortion less.

Table 3 gives part of results from 20 simulation runs, in which all data have been averaged. PCT represents $Download\_Budget/Total\_Data\_Size$, D_C represents the average distortion of the compared method, D_P represents the average distortion of proposed method, D_Re represents the average distortion reduction (D_C-D_P), P_Re represents $D\_Re/Total\_Distortion$. Fig.24 gives us more detailed and

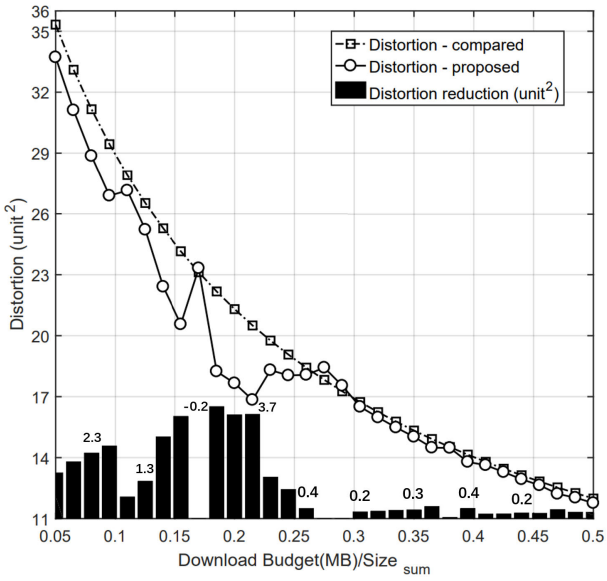

FIGURE 23. The distortion comparison between the proposed method and traditional method with the change of $Download\_Budget/Size_{sum}$.

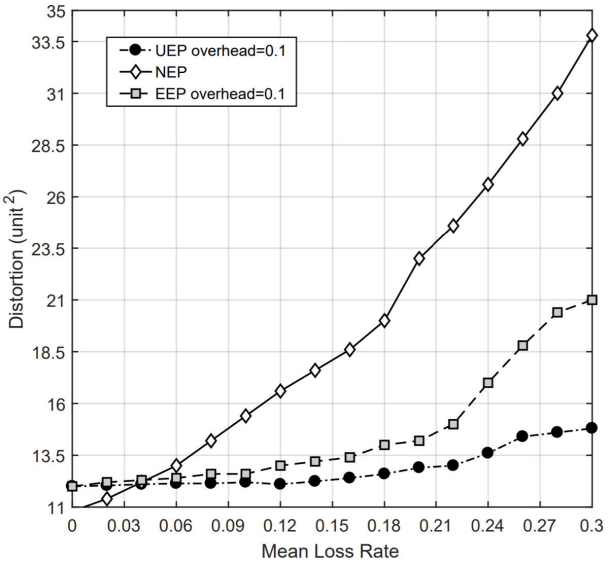

FIGURE 25. The distortion comparison between UEP, EEP and NEP at different packet loss rate.

TABLE 3. Simulation results.

Total Distortion			37.3	Total Data Size (MB)				5.26
PCT	0.05	0.095	0.155	0.2	0.26	0.29	0.41	0.5
D_C	35.3	29.4	24.2	21.3	18.4	17.3	13.8	12.0
D_P	33.6	26.9	21.5	18.1	18.0	17.1	13.6	11.8
D_Re	1.72	2.55	2.63	3.25	0.43	0.14	0.2	0.18
P_Re	4.61	6.84	7.05	8.71	1.15	0.38	0.54	0.48

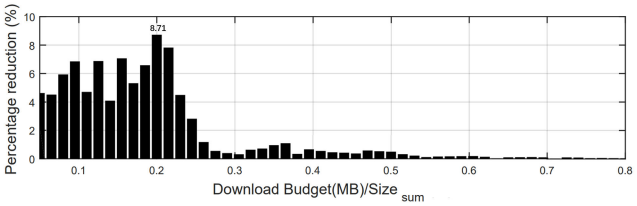

FIGURE 24. The distortion reduction ratio.

instinctive results. It is clearly depicted that, when $PCT \in (0, 0.25)$, the distortion diminution is very evident. The P_Re reach 8.71% when the PCT is 0.2. When $PCT > 0.6$, the effect is less evident.

For packet loss in lossy networks, we further compare the proposed scheme with the EEP (Equal Error Protection) and NEP (No Error Protection). We set the total transmission budget to be equal. Fig.25 gives the corresponding distortion results for different packet loss rate. As can be seen from the curves, NEP performs optimally under the lossless network due to the other two schemes' coding overhead. As the packet loss rate increases, the distortion of EEP and NEP increase rapidly, the UEP scheme is more gradual. This is because the proposed scheme provides unequal error protection for different important data, ensuring that geometry data is fully recovered with higher probability. The results show that the proposed scheme can bring less distortion especially when the network condition is poor.

VII. CONCLUSION

Streaming VEs over lossy network is a meaningful while challenging problem. We propose a three-stage progressive transmission scheme for VEs which is scalable to the network bandwidth and packet loss. In the Pre-coding Stage we firstly advance the current interest management algorithm to better fit the user's visual and behavioral characteristics. Then we bring forward a scene distortion estimation algorithm to calculate and evaluate the scene distortion effectively. On the basis of these two algorithms, we propose a scene update strategy. After that, the Transmitting Stage particularly adopt expanding window fountain (EWF) codes to provide an unequal error protection (UEP) for data with different importance. In the third After-process Stage, we employ a geometry prediction algorithm for the scenario where packet loss exists. The experimental results show that the proposed scheme can lower distortion especially when it comes to a relatively inferior network condition.

REFERENCES

[1] D. Wu, H. Shi, H. Wang, R. Wang, and H. Fang, "A feature-based learning system for Internet of Things applications," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1928–1937, Apr. 2019.

[2] D. Wu, Z. Zhang, S. Wu, J. Yang, and R. Wang, "Biologically inspired resource allocation for network slices in 5G-enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 9266–9279, Dec. 2019.

[3] P. Zhang, X. Kang, D. Wu, and R. Wang, "High-accuracy entity state prediction method based on deep belief network toward IoT search," *IEEE Wireless Commun. Lett.*, vol. 8, no. 2, pp. 492–495, Apr. 2019.

[4] D. Wu, L. Deng, H. Wang, K. Liu, and R. Wang, "Similarity aware safety multimedia data transmission mechanism for Internet of vehicles," *Future Gener. Comput. Syst.*, Vol. 99, pp. 609–623, Oct. 2019.

[5] J. Royan, P. Gioia, R. Cavagna, and C. Bouville, "Network-based visualization of 3D landscapes and city models," *IEEE Comput. Graph. Appl.*, vol. 27, no. 6, pp. 70–79, Nov./Dec. 2007.
[6] S. Kumar, J. Chhugani, C. Kim, D. Kim, A. Nguyen, P. Dubey, C. Bienia, and Y. Kim, "Second life and the new generation of virtual worlds," *Computer*, vol. 41, no. 9, pp. 46–53, Sep. 2008.
[7] C.-H. Chien, S. Hu, and J. Jiang, "Bandwidth-aware peer-to-peer 3D streaming," in *Proc. 8th Annu. Workshop Netw. Syst. Support Games (NetGames)*, Paris, France, 2009, pp. 1–6.
[8] C.-H. Chu, Y.-H. Chan, and P. H. Wu, "3D streaming based on multi-LOD models for networked collaborative design," *Comput. Ind.*, vol. 59, pp. 863–872, Dec. 2008.
[9] S.-Y. Hu, "A case for 3D streaming on peer-to-peer networks," in *Proc. ACM 11th Int. Conf. 3D Web Technol.*, 2006.
[10] S. Mondet, "Streaming of plants in distributed virtual environments," in *Proc. 16th ACM Int. Conf. Multimedia*, 2008.
[11] S. Hu, J. Jiang, and B. Chen, "Peer-to-peer 3D streaming," *IEEE Internet Comput.*, vol. 14, no. 2, pp. 54–61, Mar./Apr. 2010, doi: 10.1109/MIC.2009.98.
[12] T. Forgione, A. Carlier, G. Morin, W. T. Ooi, V. Charvillat, and P. K. Yadav, "DASH for 3D networked virtual environment," in *Proc. ACM Multimedia Conf. Multimedia Conf.*, 2018.
[13] D. Wu, Q. Liu, H. Wang, Q. Yang, and R. Wang, "Cache less for more: Exploiting cooperative video caching and delivery in D2D communications," *IEEE Trans. Multimedia*, vol. 21, no. 7, pp. 1788–1798, Jul. 2019.
[14] D. T. Ahmed, S. Shirmohammadi, J. C. de Oliveira, and J. Bonney, "Supporting large-scale networked virtual environments," in *Proc. IEEE Symp. Virtual Environ., Hum.-Comput. Inter. Meas. Syst.*, Ostuni, Italy, Jun. 2007, pp. 150–154.
[15] W. Zhi, "Network-driven dynamicscheduling algorithm for large-scale 3D scene," in *Proc. Int. Commun. Conf. Wireless Mobile Comput. (InIet)*, 2011, pp. 384–388.
[16] S.-Y. Hu, T.-H. Huang, S.-C. Chang, W.-L. Sung, J.-R. Jiang, and B.-Y. Chen, "FLoD: A framework for peer-to-peer 3D streaming," in *Proc. IEEE 27th Conf. Comput. Commun. (INFOCOM)*, Phoenix, AZ, USA, Apr. 2008, pp. 1373–1381.
[17] N. Matsumoto, Y. Kawahara, H. Morikawa, and T. Aoyama, "A scalable and low delay communication scheme for networked virtual environments," in *Proc. GlobeCom Workshops Global Telecommun. Conf. Workshops*, 2004.
[18] M. Almashor and I. Khalil, "Reducing network load in large-scale, peer-to-peer virtual environments with 3D Voronoi Diagrams," in *Proc. IEEE Int. Conf. High Perform. Comput.*, Dec. 2010, pp. 1–10.
[19] H. R. Maamar, A. Boukerche, and E. Petriu, "Streaming 3D meshes over thin mobile devices," *IEEE Wireless Commun.*, vol. 20, no. 3, pp. 136–142, Jun. 2013.
[20] Z. Li, J. Chen, and Z. Zhang, "Socially aware caching in D2D enabled fog radio access networks," *IEEE Access*, vol. 7, pp. 84293–84303, 2019.
[21] Z. Li, Y. Jiang, Y. Gao, D. Yang, and L. Sang, "On buffer-constrained throughput of a wireless-powered communication system," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 2, pp. 283–297, Feb. 2019.
[22] J. Guo, Y. Liu, L. Sang, and H. Yang, "Distortion-aware virtual environments transmission scheme over a bandwidth-limited network," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Changchun, China, Aug. 2019, pp. 671–676.
[23] D. Vukobratović, V. Stanković, D. Sejdinović, L. Stanković, and Z. Xiong, "Scalable video multicast using expanding window fountain codes," *IEEE Trans. Multimedia*, vol. 11, no. 6, pp. 1094–1104, Oct. 2009.
[24] W. Wang, J. Jia, and X. Hei, "Balance visual saliency, reusability and potential relevance for caching P2P 3D streaming contents," in *Proc. IFIP Netw. Conf.*, Brooklyn, NY, USA, 2013, pp. 1–9.
[25] H. Hoppe, "Progressive meshes," in *Proc. ACM Comput. Graph. Interact. Techn.*, 1996.
[26] C. L. Bajaj, V. Pascucci, and G. Zhuang, "Progressive compressive and transmission of arbitrary triangular meshes," in *Proc. Vis.*, San Francisco, CA, USA, 1999, pp. 307–537.
[27] J. Chim, R. W. H. Lau, H. V. Leong, and A. Si, "CyberWalk: A Web-based distributed virtual walkthrough environment," *IEEE Trans. Multimedia*, vol. 5, no. 4, pp. 503–515, Dec. 2003.
[28] C. E. Bezerra, F. R. Cecin, and C. F. R. Geyer, "A3: A novel interest management algorithm for distributed simulations of MMOGs," in *Proc. 12th IEEE/ACM Int. Symp. Distrib. Simulation Real-Time Appl.*, Vancouver, BC, Canada, Oct. 2008, pp. 35–42.
[29] H. Prendinger, R. Jain, T. Imbert, J. Oliveira, R. Li, and M. Madruga, "Evaluation of 2D and 3D interest management techniques in the distributed virtual environment DiVE," *Virtual Reality*, vol. 22, pp. 263–280, Sep. 2018.
[30] M. Aljaafreh, H. R. Maamar, and A. Boukerche, "An efficient object discovery and selection protocol in 3D streaming-based systems over thin mobile devices," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Shanghai, China, Apr. 2013, pp. 2393–2398.
[31] H. Li, M. Li, and B. Prabhakaran, "Middleware for streaming 3D progressive meshes over lossy networks," in *Proc. ACM Trans. Multimedia Comput., Commun., Appl. (TOMM)*, 2006, pp. 282–317.
[32] M. Luby, "LT codes," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Vancouver, BC, Canada, Nov. 2002, pp. 271–280.
[33] G. Al-Regib and Y. Altunbasak, "3TP: An application-layer protocol for streaming 3-D graphics," in *Proc. Int. Conf. Multimedia Expo (ICME)*, Baltimore, MD, USA, Jul. 2003, p. I-421.

YITONG LIU was born in 1982. She received the Ph.D. degree in communication engineering from the Beijing University of Posts and Telecommunications, in 2015, where she is currently a Lecturer. Her researches focus on adaptive streaming, video coding, and video quality.

JINGFENG GUO is currently pursuing the bachelor's degree with the Beijing University of Posts and Telecommunications. His research mainly concentrates on virtual reality and machine learning.

KEN DENG is currently pursuing the bachelor's degree with the Beijing University of Posts and Telecommunications. Her research mainly concentrates on virtual reality and temporal action detection.

YISHI LIU received the master's degree in communication engineering from the Beijing University of Posts and Telecommunications, in 2018. She is currently a Patent Attorney with China Patent Agent (H.K.) Ltd. Her researches mainly focus on virtual reality and 3D transmission.