Client-Driven Strategy of Large-Scale Scene Streaming

Laixiang Wen, Ning Xie, and Jinyuan Jia(✉)

Tongji University, Shanghai 201804, China
`{2010wenlaixiang,ningxie,jyjia}@tongji.edu.cn`

Abstract. Different from the strategy of virtual scene in the stand-alone application, web version may have larger scale scene and more users online at same time. However, because of the network delay (latency) and limitation of computational power of the servers, it causes that users unable to interactively access the virtual scene fluently. Meanwhile, the large scale virtual scene cannot be successfully loaded in the entry-level client machine. In this paper, we propose a client-driven strategy of scene streaming in order to solve large-scale data transmission. The experiment demonstrates that our method enables clients to enter the scene roaming and reduce the server network load. Meanwhile, it also adapts different network architectures.

Keywords: Lightweight progressive meshes (LPM) · Web3D · Scene streaming and dynamic double layer AOI (D-DLAOI)

1 Introduction

Large-scale scene transmission is one of the key parts in web-based 3D computer graphics, where 3D scene data is downloaded from servers to clients, which charge for rendering and roaming. It is widely used in many multimedia applications, such as 3D video games and virtual reality projects. Recently, the data in video games is increasing dramatically. For example, by 2014, the most famous online virtual world game, *Second Life*, has approximately 1 million users and over 270 TB data. It is impossible to publish this game through DVD disc. Also, personal computers or smart phones cannot download entire virtual world data and run the application once.

In common, these huge amount of data is usually stored on server. Once connecting established by clients, the required parts will be sent to the clients. Thus, object selection and transmission are essential operations to guarantee users' interactive operation on the virtual world. Rest of data will be transmitted by users' requests. Many client/server (C/S) based distributed virtual environment (DVE) systems work on tasks such as large-scale scene organization, priority of data transmission, adaptation of client hardware and Internet bandwidth [1,2]. Data selection and transmission towards area of interest (AOI) [3] is not efficient, because users' trajectories of the scene roaming are rather irregular.

Q. Tian et al. (Eds.): MMM 2016, Part II, LNCS 9517, pp. 93–103, 2016.
DOI: 10.1007/978-3-319-27674-8_9

Server carefully processes all requests by clients so as to make load heavily. So, it is not suitable for mobile devices. P2P technique is applied to DVE in order to reduce the load in low speed of network [4,5]. It focused on discovering the optimal download source. However, scene organization graph, priority of data transmission and configuration of terminal devices have not considered yet.

In this paper, we present a 3D scene data transmission technique to stream large scale geometry scene driven by configuration of client devices in order to realize scene roaming in real time [6]. A self-adaptive method is proposed to generate hierarchy grid structure including hierarchy scene structure and its description file. The scene organization enables to speed up the procedure of sub-scene and objects selection. The description file ensures that the selection operations can be successfully completed at the client side. Meanwhile, it guarantees the extensibility of the server so as to be a pure data server without any client information maintaining. In order to further speed up object culling, we propose a *dynamic double layer AOI (D-DLAOI)* to determine the download priority of objects. D-DLAOI has double layers and four regions. For each region, it maintains a priority queue. According to density of object in D-DLAOI, the size of the inner layer is adjusted dynamically so as to control the number of request and maintain the sent request to be up-to-date. It makes sure the most needed objects will be transmitted at first (the current visible objects) and culls the most potential objects (the potential visible objects). Finally, lightweight progressive mesh (LPM) [7] is applied to achieve the stream transmission without the entire scene file under the optimal resolution.

We demonstrate our idea using various large scale scene examples, and present the comparison to evaluate the usability and efficiency of our tool.

2 Related Works

Communication Framework. Communication framework defines how computers organize together as a network. The common schemes are Client/Server (C/S), Peer-to-Peer (P2P) and Hybrid frameworks [8]. C/S is adopted by many multimedia 3D applications. For example, massively multiplayer online games (MMOGs) are built on C/S such as *ActiveWorlds*, *Second Life*, and *World of Warcraft*. This structure ensures data security and consistency. P2P media streaming has been significant progress in recent years [9]. Technically, P2P is different from C/S from the aspects of access pattern and task distribution. But, P2P data transmission is same with C/S framework, although for only small amount of data. To improve data persistence, hybrid framework is proposed to add server into P2P framework [10].

Grid Generation in 3D VE. Grid is one of the important structure for many 3D computer graphics problems such as ray tracing, collision detection and path finding. Comparing with Octree and kd-tree, 3D uniform grid is suitable for the dynamic scene. However, the computational efficiency highly depends on the grid creation strategy. Currently, the *cost evaluation* method and the *empirical model* method are two most efficient strategies. The cost evaluation method needs the classification step in advance under certain assumption [11]. The empirical model is much easier because it is based on the experience and statistics [12,13].

3D Visual Component Selection and Streaming. Visual scene streaming can be summarized into *3D geometry scene streaming (GSS)* and *image-based scene streaming (ISS)*. ISS assumes that clients do not have enough ability for 3D scene rendering. Instead, server renders and sends the rendered image frames to clients. However, when the number of users increases, processing becomes immeasurable that server has to render many different copies of contents according to many different clients' viewpoints. Applications are such as *CloudGame* [14] and *GamingAnyWhere* [15].

Alternatively, in GSS, server sends 3D geometries instead of rendered videos to clients, who render the received primitives as a visual scene. It is widely used in DVE system, where only currently visible geometries will be transmitted to clients. Wang et al. [16] applied the area of interest (AOI) to judge object visibility in local region. Constructive solid geometry (CSG) is used to construct the objects in *SecondLife*. When geometry component are contained by several visible objects, it will be sent at first as priority. By this way, the procedure of scene construction can be accelerated. Furthermore, in order to balance transmission speed and 3D model resolution, Level of detail (LOD) technique is introduced to generate 3D objects under multiple resolutions [8]. Models under low resolution will be transmitted first so as to ensure the speed of scene construction. It is simple and straightforward. However, LOD method increases Internet bandwidth consumption, because of redundancy that same object have multiple resolutions. Progressive mesh (PM) [17] is introduced to avoid this problem in P2P transmission.

The challenge is how to enhance the user experience on the visible interaction. All techniques endeavor to reduce the latency during 3D scene streaming including rapid object culling, transmission priority, resolution controlling are needed to be investigated.

3 Overview

This paper introduces a 3D scene transmission pipeline to stream large scale virtual scene which is adaptive to the client configuration. Our approach consists of three main steps:

- **Multiple-resolution 3D space adaptive grid creation.** According to the density of the objects in the scene, multi-level grids are recursively calculated in order to generate the hierarchical structure. It helps to speed up the process of object culling. Meanwhile, scene description file is also generated for client-oriented data request.
- **Visible scene determination.** D-DLAOI is proposed to determine the download priority of objects. D-DLAOI has double layers (the internal layer and the external layer). In order to avoid excessive requests by clients, the internal layer can be real-time justified according to the density of the objects.

D-DLAOI covers the user visible or potential visible region. Only objects in D-DLAOI are the candidates for transiting. According to the user's view frustum, D-DLAOI are further divided into four regions with different predefined priorities including a visible region, two potential visible regions, and a pre-download region. In a particular region, the priority of object is determined by the distance to the viewpoint, and the included angle with the sight line.
- **Object resolution optimization.** After applying LPM for streaming, we need to calculate the optimal resolution for visible objects based on human visual mechanism, the configuration of the clients and network environment.

By this pipeline, server do not have to maintain any information on the client devices anymore and focuses on the operation of VE data. Hence, coupling degree is significantly reduced at server.

4 Multiple-resolution 3D Space Adaptive Grid Creation

We adaptively judge the mesh resolution from two aspects including the size of the scene and the density of objects in the scene.

In this paper, we give the method to calculate the resolution of the 2D mesh in one layer as below:

$$R_i = \lambda L_i \sqrt{\frac{N}{A}}, \qquad (i \in \{\mathrm{X},\ \mathrm{Y}\}, 0 < \lambda \leq 1), \tag{1}$$

where R_i is the resolution in the projected 2D for certain axis, i (X or Y in our system); λ is a parameter; N is the number of the objects in the given scene; A is the projected area of the bounding box of 3D grid; L_i is the length of the certain corresponding axis. In practice, bounding box of scene can be any rectangular parallelepiped but not just the cube (or regular hexahedron) with equal edge lengths. In worse case, the bad resolution will result in lower computational efficiency. In order to maintain the global optimal grid division, we need to justify the grid resolution in certain axis. In detail, once some grid resolution is less than 1, we set up it to 1, while the other is set to $\lambda\sqrt{N}$.

To further accelerate the process of object culling, we propose a hierarchical grid structure. First, we generate a one layer uniform grid for a scene who contains N objects. Then, we traverse every cell of grid to check whether it can be further divided. If so, it will be divided. This process will go on recursively until the condition is not satisfied, where the number of object in the certain cell is less than the threshold.

In practice, a large scale scene is organized by a series of sub scenes. It is rare to generate a global uniform grid. Especially in MMOGs, the grids are usually designed by scene designers. Alternatively, the algorithm processes the grid generation for each sub-scene.

Then, lightweight process is necessary to remove redundant objects in the scene. We apply LPM [7] on each non-redundant object afterwards. The hierarchical information of a scene is illustrated as Fig. 1. In this structure, there are three kinds of nodes: root node, intermediate node and leaf node.

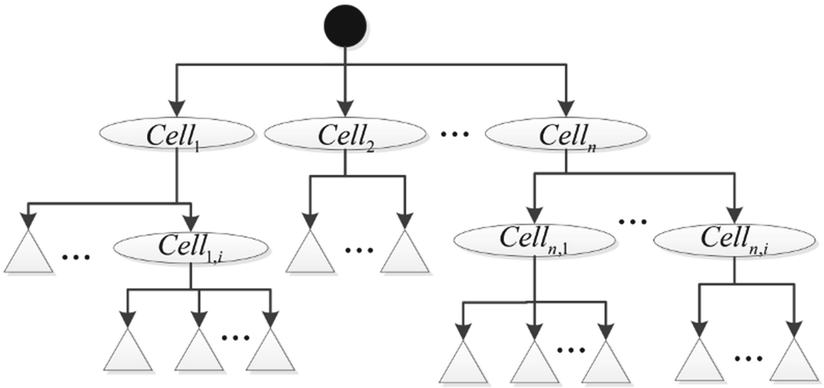


Fig. 1. Scene hierarchical structure.

- **Root node.** It is also called *scene root*. It has the highest level information including the sub-scene division, the configuration of each sub-scene (position and size), and ID of sub-scenes.
- **Intermediate node.** It is the node for a sub-scene. There are object data (position, size, pose and its ID) and the information of nested sub-scene.
- **Leaf node.** It is the object node including ID of objects and its configuration. Note that LPM can be obtained using its object ID.

Note that it has three significant characters including hierarchical structure, lightweight processing and streaming transmission. Nodes contain less amount of data, because it only has the scene descriptor file, without the object data. We later send these compact files to clients in order to assemble scene.

5 Scene Streaming Assemble Strategy

Once receiving the descriptor files for multi-levels, client enables to judge the visible scene, the priority of the object transmission, and the resolution of LOD of objects. All these are called *scene streaming assemble*. The aim is to accelerate the network transmission but not scene rendering. Therefore, AOI is applied rather than object culling, back culling, and occlusion culling.

5.1 Dynamic Double Layer AOI (D-DLAOI)

For online game, *area of interest (AOI)* is a technique to reduce communication burden even though most of the cases all-to-all communication is required within an AOI [18]. Say that AOI is the area that surrounds the player or avatar in the center. In very large scene roaming, visible objects are only who are covered by AOI in order to reduce real time computational cost of object visibility. The classic AOI is in disc shape, while the human cones of vision is sector. *Double* *layer AOI (DLAOI)* is then proposed to further simulate on this issue [19]. The advantage of DLAOI is to preload data in advance for rendering in order to accelerate the rendering. However, it still can not solve the issue on dense scene where contains large numbers of objects.

We propose a dynamic DLAOI (D-DLAOI) to dynamically change the size of the inner area according to the density of the objects as illustrated in Fig. 2. Similar to DLAOI, our D-DLAOI also has four partitions. As illustrated in Fig. 2(a), Q_{1a} is the *visible area*. Q_{1b} and Q_{2a} are also *potential areas*, which means that the inside objects can be observed in the next time slot. Q_{2b} is the pre-loaded area which will be loaded once no user's action exists after other three areas have been loaded. Area priority order is $Q_{1a} > Q_{1b} > Q_{2a} > Q_{2b}$. In client, once scene description file is received, the hierarchical structure will be created as illustrated in Fig. 2(b). In practice, the density of objects in the scene is not uniform. To further improve performance, we can also dynamically justify the size of Q_{1b}. In detail, we reduce the size of Q_{1b}, when $p < \varepsilon_{sh}$. Alternatively, the size of Q_{1b}, when $p > \varepsilon_{in}$, where

$$p = 1 - \frac{N_{\text{inner}}}{N_{\text{AOI}}}; \tag{2}$$

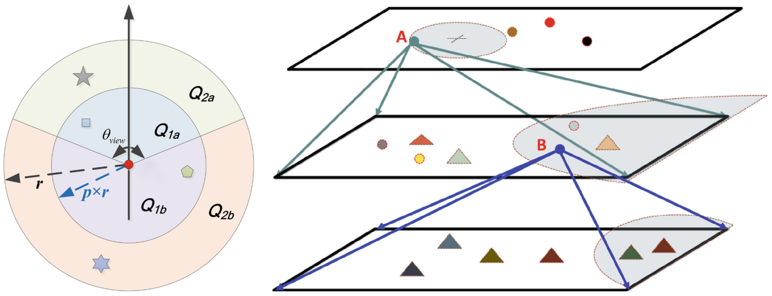
N_{inner} is the number of the objects in the inner region of D-DLAOI; N_{AOI} is the number of the objects in AOI. ε_{sh} and ε_{in} are thresholds ($\varepsilon_{sh} = 0.01, \varepsilon_{in} = 0.99$ in our experiment).

5.2 Object Priority Determination and LOD Resolution

For designing the priority determination and LOD resolution, we follow the two common assumptions of the human vision below:

- For the viewpoint, closer to the target object;
- For the sight line, less offset from the sight line.

Therefore, it may not be helpful to improve the visual impact by loading the entire model data, whereas it will increase the amount of data transmission. By



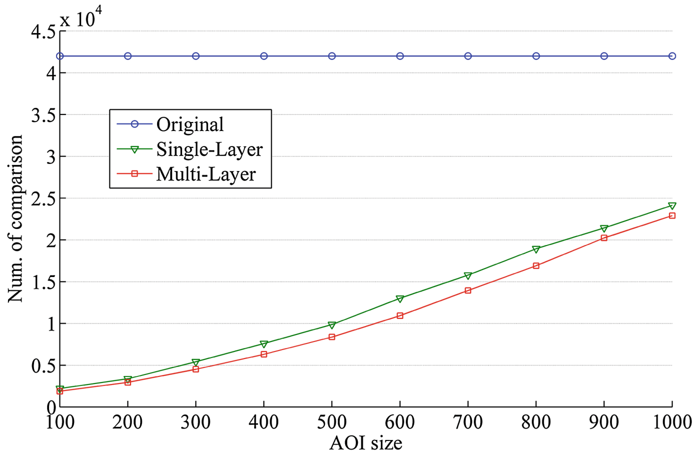
(a) D-DLAOI (b) Hierarchical structure

Fig. 2. D-DLAOI.

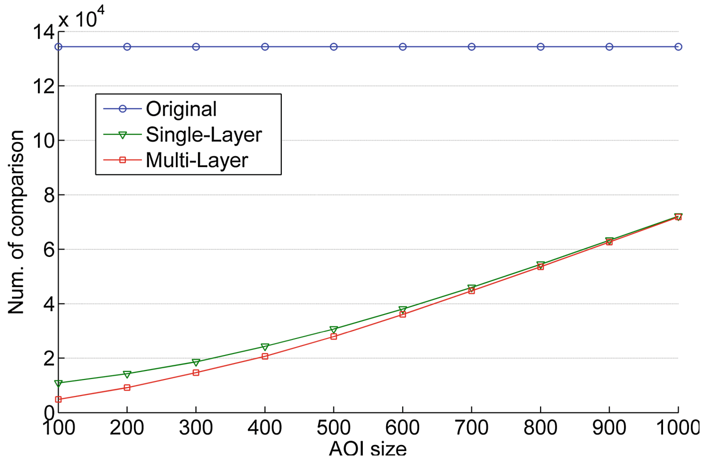
doing so, users have to wait longer for scene roaming. LPM [7] is a powerful tool to distribute the optimal amount of 3D model data, p^* according to needs of LOD as below:

$$p^* = p_{\mathrm{m}}(\gamma b + \delta r + \alpha(1 - a) + \beta(1 - d)), \tag{3}$$

where p_{m} is the maximum number of grid; b, r, a, and d are the normalized parameters. In detail, b means the network bandwidth; r means the rendering capabilities of the client; a is the offset angle from sight line; d is the distance from the view point to the target object. α, β, γ and δ are the coefficients ($\alpha + \beta + \gamma + \delta = 1$).



(a) size = 1000, N = 2000



(b) size = 2000, N = 6400

Fig. 3. D-DLAOI evaluation in two scenes.

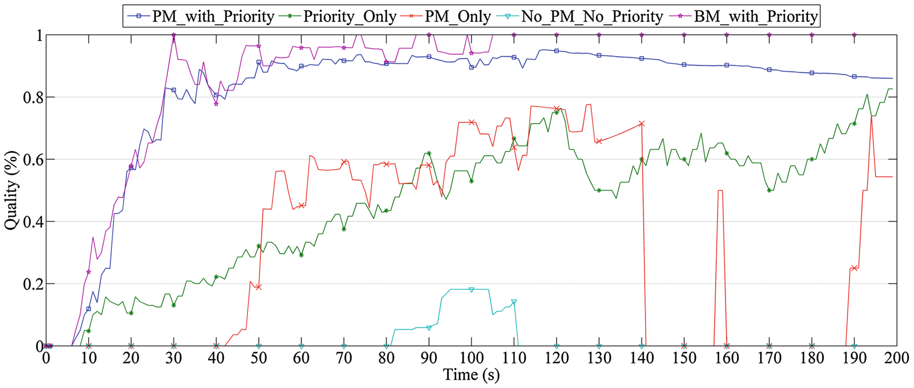


Fig. 4. The comparison on the human visual impact.

6 Experimental Results

In this section, we experimentally demonstrate the usefulness of our proposed method.

First, we evaluate our proposed D-DLAOI method through an illustrative experiment. We randomly generate two scenes in different sizes. One has 2000 objects with the size in 1000 × 1000. The other has 6400 objects with the size in 2000 × 2000. We compare three methods including the original AOI, the single layer dynamic AOI and our proposed D-DLAOI. All of them follow the same roaming trajectory. For the original AOI and the single layer dynamic AOI, the size of grid is in 100 × 100. For our D-DLAOI method, the shortest length of the longest axis is set to 100. In Fig. 3, we show the comparison on the number of visible object detection. Our D-DLAOI method is the best in three methods. From Fig. 3(b), with the increasing of the size of AOI, the performance of D-DLAOI is closer to the single layer method. But they are still much better than the original AOI.

Next, we evaluate quality of human visual impact. In this experiment, the network is under 10 Mbps, the size of scene is in 4000 × 4000 with 5000 objects, which are Stanford Bunny chosen for simple. The roaming trajectory is set as circle ($(x - x_c)^2 + (y - y_c)^2 = R^2$), where the center, (x_c, y_c) is $(2000, 2000)$; the radius, R is 1000; and the speed is $2m/s$. The measurement of the quality of human visual impact is defined as

$$\text{Quality} = \sum p'_i / \sum p_i, \tag{4}$$

where p_i is the amount of data needed for the ith object; p'_i is the received data for the ith object at this moment. The follow five methods are compared:

- **PM_with_Priority** is our proposed method in this paper;
- **Priority_Only** is to transmit the original scene data with priority;

- **PM_Only** is applied PM transmission without priority;
- **No_PM_No_Priority** is to transmit the original scene data without priority;
- **BM_with_Priority** is transmit the base mesh (BM) of the scene with priority.

The experimental results are plotted in Fig. 4, showing the human visual impact among five methods. PM_with_Priority performs the best, demonstrating that our proposed method is promising. For more realistic experiment, we also run demonstrations on the scene of forest and city as in Figs. 5 and 6.

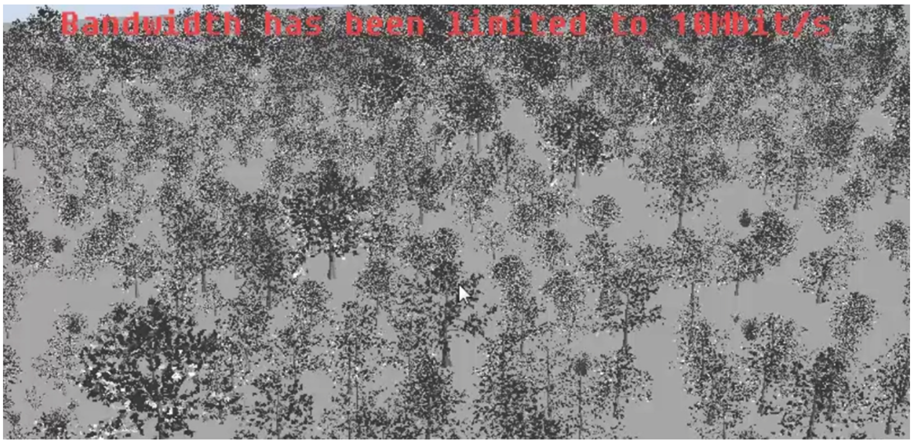


Fig. 5. Demo of forest scene roaming.

Fig. 6. Demo of city scene roaming.

7 Conclusion and Future Work

We present a client-driven strategy of scene streaming so as to achieve real time large-scale 3D scene roaming. Our main contributions in this paper are (i) we propose a self-adaptive method to generate hierarchy grid structure including hierarchy scene structure and its description file; (ii) we propose a D-DLAOI to further speed up scene culling and object selection; (iii) we apply LPM in order to improve the human visual impact in the large scale scene roaming. The experiments demonstrated the effectiveness of our proposed approach in the realistic forest and city roaming applications.

In the future, our plan will focus on improving the performance of multiple mobile VE clients for the collaborative works. Specifically, our aim is to Accurate localization for the optimal downloading source [20] in P2P-DVE.

Acknowledgments. The research was supported jointly by National Science Foundation of China (No. 61272276), National Twelfth Five-year Plan Major Science and Technology Project (No. 2012BAC11B00-04-03), Special Research Fund of Higher College Doctorate (No. 20130072110035), Key Science and Technology Project of Jilin (No. 20140204088GX) and Tongji University Young Scholar Plan (No. 2100219052).

References

1. Hu, S.Y.: A case for 3D streaming on peer-to-peer networks. In: Proceedings of Web3D 2006, pp. 57–63 (2006)
2. Li, F.W.B., Lau, R.W.H., Kilis, D., Li, L.W.F.: Game-on-demand: an online game engine based on geometry streaming. ACM Trans. Multimedia Comput. Commun. Appl. **7**, 1–22 (2011)
3. Wang, W., Jia, J.: An incremental smlaoi algorithm for progressive downloading large scale webvr scenes. In: Proceedings of the 14th International Conference on 3D Web Technology, pp. 55–60 (2009)
4. Esch, M., Botev, J., Schloss, H., Scholtes, I.: P2P-based avatar interaction in massive multiuser virtual environments. In: International Conference on Complex, Intelligent and Software Intensive Systems, CISIS 2009, pp. 977–982 (2009)
5. Gurler, C.G., Tekalp, M.: Peer-to-peer system design for adaptive 3D video streaming. IEEE Trans. Commun. Mag. **51**, 108–114 (2013)
6. Gulliver, S.R., Ghinea, G.: Defining user perception of distributed multimedia quality. ACM Trans. Multimedia Comput. Commun. Appl. (TOMM) **2**(4), 241–257 (2006)
7. Wen, L.X., Jia, J.Y., Liang, S.: Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 95–103 (2014)
8. Cavagna, R., Royan, J., Gioia, P., Bouville, C., Abdallah, M., Buyukkaya, E.: Signal processing: image communication. Special Issue on Advances in Three-dimensional Television and Video **24**(1–2), 115–121 (2009)
9. Hu, S.Y., Huang, T.H., Chang, S.C., Sung, W.L., Jiang, J.R., Chen, B.Y.: Flod: a framework for peer-to-peer 3d streaming. In: The 27th IEEE Conference on Computer Communications (INFOCOM) (2008)

10. Botev, J., Hohfeld, A., Schloss, H., Scholtes, I., Sturm, P., Esch, M.: The HyperVerse: concepts for a federated and Torrent-based '3D Web'. Int. J. Adv. Media Commun. **2**(4), 331–350 (2008)
11. Ize, T., Shirley, P., Parker, S.: Grid creation strategies for efficient ray tracing. In: IEEE Symposium on Interactive Ray Tracing, pp. 27–32 (2007)
12. Wald, I., Ize, T., Kensler, A., Knoll, A., Steven, G.: Ray tracing animated scenes using coherent grid traversal. ACM Trans. Graph. (TOG) **25**(3), 485–493 (2006)
13. Lagae, A., Dutre, P.: Compact, fast and robust grids for ray tracing. Comput. Graph. Forum **27**(4), 1235–1244 (2008)
14. Chuah, S.P., Yuen, C., Cheung, N.M.: Cloud gaming: a green solution to massive multiplayer online games. IEEE Trans. Wirel. Commun. **21**(4), 78–87 (2014)
15. Huang, C.Y., Hsu, C.H., Chang, Y.C., Chen, K.T.: GamingAnywhere: an open cloud gaming system. In: Proceedings of the 4th ACM Multimedia Systems Conference, pp. 36–47 (2013)
16. Wang, M.F., Jia, J., Zhongchu, Y.X., Zhang, C.X.: Organizing neighbors self-adaptively based on avatar interest for transmitting huge DVE scenes. In: Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, pp. 105–111 (2014)
17. Hoppe, H.: Progressive meshes. In: Proceedings of the 23rd Annual ACM Conference on Computer Graphics and Interactive Techniques, pp. 99–108 (1996)
18. Ahmed, D.T., Shirmohammadi, S.: An auxiliary area of interest management for synchronization and load regulation in zonal P2P MMOGs. In: IEEE International Workshop on Haptic Audio visual Environments and Games, pp. 36–41 (2008)
19. Rooney, S., Bauer, D., Deydier, R.: A federated peer-to-peer network game architecture. IEEE Commun. Mag. **42**(5), 114–122 (2004)
20. Liu, E.S., Theodoropoulos, G.K.: Interest management for distributed virtual environments: a survey. ACM Comput. Surv. **46**(4), 51:1–51:42 (2014)