# WebTorrent Based Fine-grained P2P Transmission of Large-scale WebVR Indoor Scenes

Yonghao Hu
School of Software Engineering
Tongji University
China
foreverhyh58@hotmail.com

Zhaohui Chen
School of Software Engineering
Tongji University
China
bluefchen@gmail.com

Xiaojun Liu
School of Software Engineering
Tongji University
China
xjliu1204@126.com

Fei Huang
School of Software Engineering
Tongji University
China
huangfei@tongji.edu.cn

Jinyuan Jia
School of Software Engineering
Tongji University
China
jyjia@tongji.edu.cn

## ABSTRACT

The latency of transmitting large-scale[1] WebVR scenes over mobile Internet is known as the bottleneck problem. This paper attacks this challenging problem by combining graphics based progressive transmission and networking based P2P transmission together. Different with those pure research DVE (Distributed Virtual Environment)-P2P works built on simulation platform, a novel WebVR-P2P framework is realized based on WebTorrent and WebGL. At server side, huge WebVR scenes are lightweighted by finding all repetitive components and removing redundant ones, that avoids unnecessary transmission. Furthermore, large-scale WebVR indoor scenes are divided into smaller fine-grained subspaces in terms of closeness and visibility to lower networking congestions. These two preprocessing steps are integrated to decrease less bandwidth occupation at utmost. Then, each fine-grained subspace is packaged adaptively in terms of Frustum Fill Ratio (FFR) for smooth and efficient transmission. A new WebTorrent framework is extended to transmit Web3D files and all packaged WebVR subspaces are transferred in the peer-to-peer style. At Web-end, two-thread, one for package transferring and the other for rendering, is employed asynchronously to realize online real time rendering. Finally, WebVR-P2P platform with three layer architecture is implemented based on all above key technologies, a large-scale WebVR Metro scene (about 1GB) is chosen to test for P2P transmission performance in this WebVR-P2P platform, the practical experimenting results are conducted to show the effectiveness and potentiality of our proposed solution.

## CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; *Redundancy*; Robotics • **Networks** → Network reliability

## KEYWORDS

WebRTC, WebTorrent, WebVR, Fine-grained Preprocessing, Lightweight Preprocessing; Frustum Fill Ratio (FFR), Progressive Transmission

**ACM Reference format:**

Y. Hu, C. Chen, X. Liu, F. Huang, J. Jia. 2017. SIG Proceedings Paper in word Format. In *Proceedings of the 22th International Conference on Web 3D Technology, QLD, Australia, June 2017 (Web3D'17)*, 8 pages.
DOI: http://dx.doi.org/10.1145/3055624.3075944

## 1 INTRODUCTION

When Internet+ meets VR+, population of WebVR has been accelerated again. We are seeing the explosion of **WebVR**. Now **WebGL** has been widely supported by default in modern browsers, tools such as **Xml3D** [Xml3d 2017], **X3DOM** [X3dom 2017], **Cobweb** [Cobweb 2017]**, Three.js** [Three.js 2017]**, GlTF** [Gltf 2017]**, and A-Frame VR** [A-frame vr 2017] are allowing nearly anyone to create Web3D contents. Commercial game engines such as Unity and Unreal are starting to offer ways to export and publish directly to **Web3D**.

However, neither DVE nor WebVR has been blocked by slow downloading and rendering large scale virtual scenes over mobile Internet. In our long-term practices of WebVR R&D, we do often encounter the following problems:

(1) The amount of WebVR data required to download from server is too huge to transmit for currently available networking bandwidth to reach real time transmission.

[1] We describe large-scale or huge scene which means the total data size is over 1GB

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.
Web3D '17, June 05-07, 2017, Brisbane, QLD, Australia
© 2017 ACM. ISBN 978-1-4503-4955-0/17/06...$15.00
DOI: http://dx.doi.org/10.1145/3055624.3075944

(2) CG based WebVR progressive transmission still cannot guarantee smooth walkthrough of WebVR scenes, the latency will become rather apparent, especially encountering into such huge buildings as skyscraper or metro stations.

(3) Even though huge WebVR scenes have been downloaded successfully, it is still a difficult task to render them in real time on Web Browsers due to weak computing and limited caching capability of Web Browsers.

(4) The users' tolerance of initial visualization from downloading to rendering is becoming shorter and shorter due to web rendering inefficiency. Only within 5 seconds huge WebVR scenes should be visualized once user opening its Webpages online.

# 2 RELATED WORK

## 2.1 Lightweight Peprocessing of WebVR Scenes

The earliest idea of finding repetitive components in huge 3D models was proposed in [Shikhare d 2001]. CAI employed CPCA to find more repetitive 3D components and obtained higher compression rate in [Laixiang Wen 2015]. Furthermore, WEN proposed voxelization based finding repetitive 3D components with higher repetitiveness rate than [Laixiang Wen 2015] and stream those non-repeated 3D components with pregressive mesh (PM) [Laixiang Wen 2015]. Michael presented a framework of lightweight progressive meshing and lightweight progressive texturing in [Michael englert 2015]. LIU enhanced WEN's method by semantic lightweighting and segmenting a huge WebVR scene into some fine-grained blocks to suit Internet transmission [Xiaojun liu 2016].

## 2.2 Peer-to-Peer Transmission of WebVR:

Peer-to-peer networking is regarded as a potential solution to DVE/WebVR transmission bottleneck problem. There are rather research publications to address this hot topic and good survey can be seen in [Buyukkaya 2013] and [Yahyavi 2013]. However, it is very difficult to combine progressive transmission mechanism from graphical methodology with P2P transmission from networking methodology very well. Almost all P2P-DVE researches were experimented on simulation platform. HU developed a good simulation platform FloD conducted a good neighboring policy of DVE-P2P in [Hu s y 2008], JIA extended it into iFLoD by introducing incremental AOI [Jia jy 2014], WANG proposed the interests driven P2P neighboring and further extended iFLoD into iiFLoD [Mingfei wang 2015]. But, this P2P-DVE simulation platform is far away from real VR scenes and Internet communications, thereby, these resulted researches are lack of practicality and convincing.

## 2.3 WebRTC/WebTorrent based P2P Transmission

The emerging WebRTC technology [Webrtc 2017] provides peer-to-peer connections among Web browsers. WebTorrent is a ripe framework of WebRTC for audio and video transmission [Feross aboukhadijeh 2017; Lopez fernandez 2013; Kiran kumar guduru 2015]. To our best knowledge, there is no WebTorrent applied to P2P transmission of large-scale WebVR, Andrioti [Haroula andrioti 2015] integrated WebRTC and X3DOM together to build a collaborative 3D modeling environments but not for P2P transmission of large-scale WebVR world. Furthermore, Andrioti proposed a framework of WebVR P2P transmission using WebRTC, but there are no experimental results to show about P2P transmission over mobile Internet.

Our contributions consists in (1) a new fine-grained preprocessing procedure is provided for huge WebVR scenes at server side; (2) a new WebVR packaging mechanism is presented to suit Internet bandwidth; (3) more important, a new WebTorrent-P2P transmission framework is established and real experiments of WebVR-P2P transferring is tested on mobile Internet.

# 3 KEY TECHNOLOGIES

To solve the four challenging problems of WebVR above, accordingly, four key technologies are presented to address them as follows:

(1) At server side, huge WebVR scenes are lightweighted by finding all the repetitive components and removing redundant ones, that avoids unnecessary transmission.
(2) Huge 3D models are segmented into some smaller fine-grained blocks in terms of closeness and visibility to avoid sudden latency once encountering into heavy 3D models during walkthrough relieve networking burdens. These two preprocessing steps are integrated to decrease less bandwidth occupation at utmost.
(3) Each fine-grained block is packaged adaptively in terms of Frustum Fill Ratio (FFR) for smooth and efficient transmission.
(4) A new WebTorrent framework is extended to transmit Web3D files and all packaged WebVR blocks are transferred in the peer-to-peer style.
(5) At Web side, a double-thread, one for transferring packages and the other for rendering, is employed asynchronously to reach real time WebVR rendering.

The entire technical roadmap is illustrated as follows.

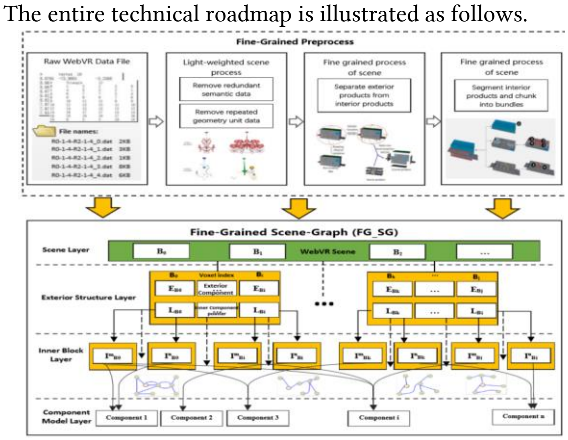



**Figure 1: Technical roadmap**

### 3.1 Lightweight Preprocessing

Lightweighting of WebVR scenes consists of semantic analysis, geometric repetitive detection and redundancy removal. For the huge 3D model without repetitive components, its exterior should be extracted from interiors by the voxelization scanning method in [Xiaojun liu 2016].

### 3.2 Fine-grained Preprocessing

Fine-grained process is designed to decompose large-scale 3D scenes that cannot be downloaded in real time into some sub-spaces within the scope of Web3D real time rendering. Each subspace is relatively self-closed space as an independent unit of downloading and rendering when roaming into this sub-space. Web browsers only download the corresponding sub-space according to the position and orientation of current viewpoint. For example, Chegongmiao Subway Station with 30,000 entities, as shown in Figure 1, cannot be rendered at once by Web Browsers and divided into fine-grained into some sub-spaces. Each one contains only 1,000 components which can be rendered in real time due to its relative space closure.

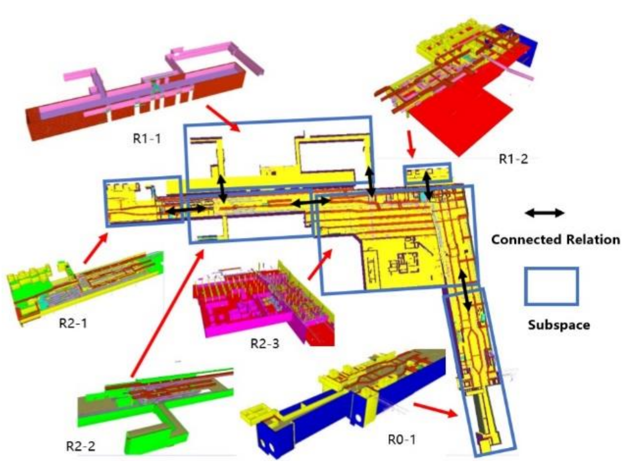



**Figure 2: Chegongmiao subway station structure and its subspaces after fine-grained preprocessing**

### 3.3 Frustum Fill Ratio (FFR)

After fine-grained preprocessing, it is necessary to sort all the 3D entities or components for each subspace according to their different visual contribution. Under this circumstances, we define a new concept, Frustum Fill Ratio (FFR), is defined to measure the proportion of the entities or components in the frustum of viewpoint (FOV).

*Definition 1.* Frustum Fill Ratio: The ratio of the volume of AABB of an entity to its file size.

The formulae can be given as follows:

$$FFR(Entity_i) = \frac{Volume(AABB(Entity_i))}{Size(Entity_i)} \qquad (1)$$

As shown in Figure 3, there are door and ticketing machine within FOV, the AABB of door is much larger than the AABB of ticket machine, however, the file size of door is much less than the ticket machine, thus, the FFR of door is higher than the ticket machine, so the door will be downloaded and rendered earlier than the ticket machine.

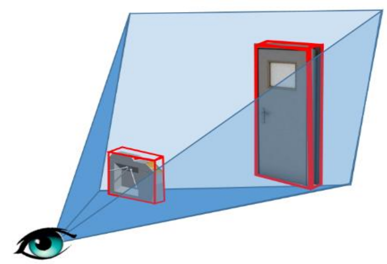


**Figure 3: Frustum Fill Ratio in FOV**

In general, the FFR of wall or floor is higher than pipes, windows and doors and other components in the building scene. The higher FFRs those components, the easier users to perceive, the earlier their downloading and rendering with higher priority. That will let users to wait as less as possible. As shown below, from the current view perspective, the higher FFR of walls and floors, the earlier their downloading and rendering, the later the water pipes, doors and windows due to their lower FFR.

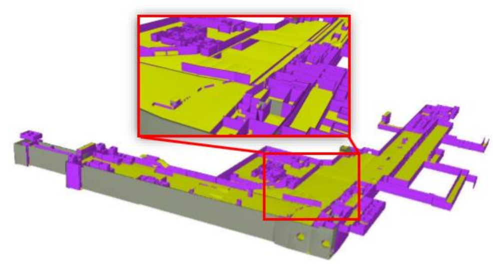


**Figure 4: The floors and walls with higher FFR are downloaded and rendered with higher priority**

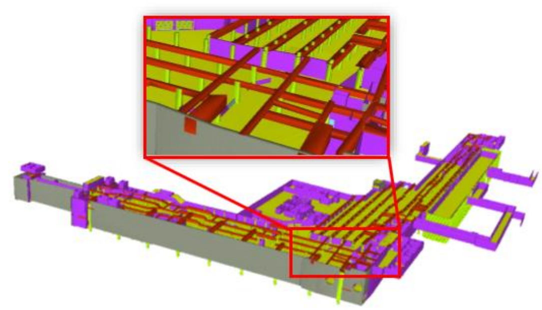


**Figure 5: The doors and columns with middle FFR are downloaded and displayed next**

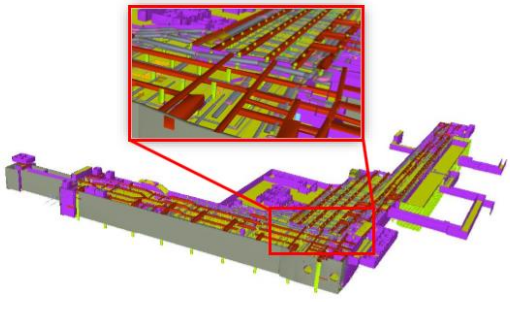


**Figure 6: Pipelines with the lowest FFR are downloaded and rendered finally**

## 3.4 Adaptive packaging

After computing FFR, all 3D entities in the subspace should be packed in terms of their FFRs for more efficient networking transmission. A package is treated as a unit of Internet transmission. Packing procedure is specified as follows:

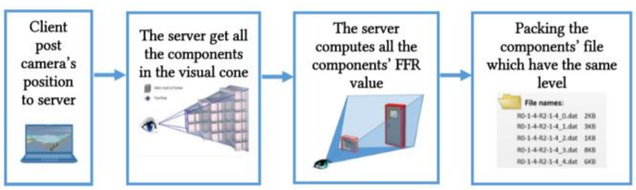



**Figure 7: Flowchart of adaptive packaging**

For example, a fine-grained sub-space (#R1-2) is shown in Figure 6, and correspondingly is packaged into the folder of 1805 components, which is stored at the server side.

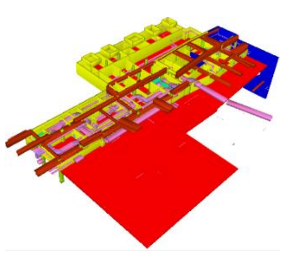


**Figure 8: The sub-space R1-2**

The packaging procedure, as depicted in Figure 7, sorts all the entities into the list as shown in Table 1 according to their FFR. Then, it packages these sequentially into one folder after another.

**Table 1: FFR list of all the entities in subspace R1-2**

| Type | AABB ( $m^2$ ) | Data Size (kb) | FFR | Priority |
|---|---|---|---|---|
| Slab | 90 | 0.987 | 91.2 | 1 |
| Covering | 85 | 1.02 | 83.3 | 2 |
| Wall | 72 | 1.93 | 37.3 | 3 |
| Beam | 64 | 2.68 | 23.9 | 4 |
| Column | 60 | 2.86 | 20.9 | 5 |
| Stair Flight | 52 | 4.02 | 13 | 6 |
| Window | 55 | 4.48 | 12.3 | 7 |
| Stair | 50 | 5.99 | 8.4 | 8 |
| Railing | 42 | 6.15 | 6.8 | 9 |
| Door | 38 | 6.34 | 5.9 | 10 |
| Flow Segment | 36 | 6.84 | 5.2 | 11 |
| Flow Fitting | 24 | 7.99 | 3 | 12 |
| Flow Controller | 19 | 8.78 | 2.16 | 13 |
| Flow Terminal | 8 | 9.86 | 0.8 | 14 |
| Building Element | 3 | 10.6 | 0.3 | 15 |

## 3.5 WebTorrent Based Web3D P2P Transmission

### 3.5.1 *Web3DTorrent Framework.*

After suitable packing, a WebTorrent based WebVR-P2P transmission framework can be established as shown in Figure 9. It consists of three modules, (1) the fine-grained preprocessing of WebVR scenes, (2) logical P2P grouping from social network, and (3) Physical P2P transmission by the aid of WebTorrent. This WebVR-P2P framework is centered on WebRTC, a Web peer-to-peer network transmission protocol.

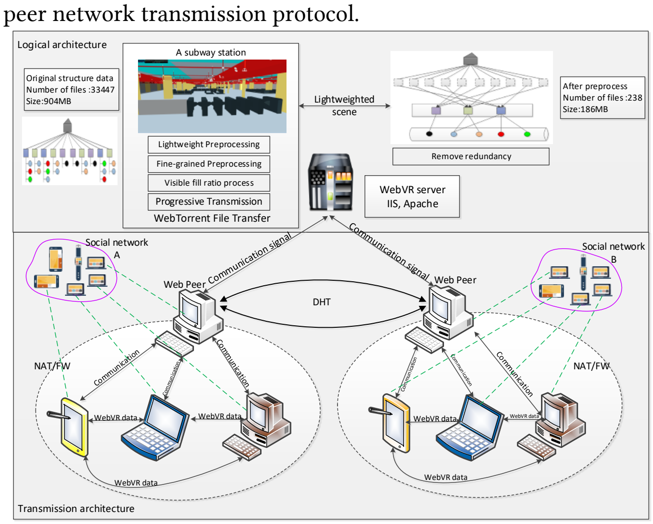



**Figure 9: WebVR-P2P Transmission Framework**

### 3.5.2 *WebVR-P2P Transmission Architecture.*

There are three layers in our proposed WebVR-P2P architecture, social network layer, logical P2P overlay network and physical P2P transmission layer.

The first layer is the social network layer to form a common user group who might be interested to share the same WebVR scene from Facebook, Twitter, WeChat and Email List. These group members may have similar roaming or interacting behaviors in the same WebVR scene through mobile Internet due to the same tendency or similar preferences. They are suitable to become the downloading service providers for others. Thus, those WebVR models downloaded by one peer are not only used for rendering itself, but can be provided for the other neighbors to download without accessing servers.

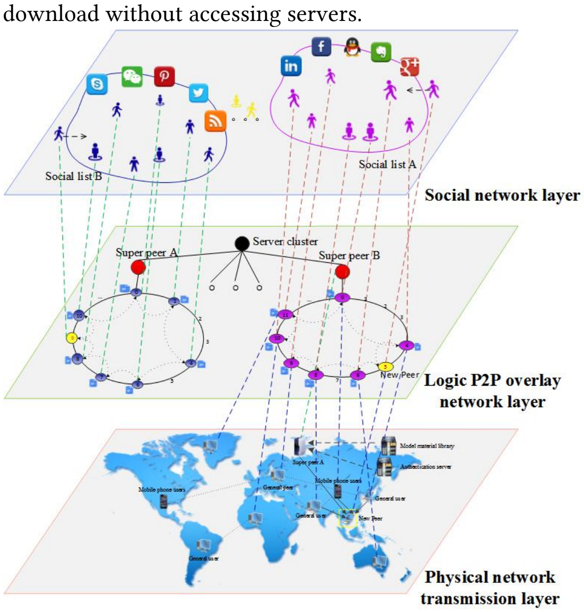



**Figure 10: WebTorrent Based WebVR-P2P architecture**

The logic P2P overlay network layer is built by the same social network group to prepare for Web3DTorrent. When a new peer requests WebVR scenes, WebTorrent quickly determines which peer stores the requested scene data by seeking distributed hash table (DHT) or querying a tracker server. The logic P2P overlay network layer maps a member of social network layer to the node of physical network transmission layer correspondingly.

### *3.5.3 Web3DTorrent P2P Transmission Flowchart.*

The peer-to-peer transferring of WebVR scenes is facilitated by two fundamental technologies.

The first one is Web Real-Time Protocol (WebRTC). WebRTC allows P2P DataChannel protocol to exchange raw data among Web browsers in real-time, which relies on Interactive Connectivity Establishment (ICE), it can traverse network address translators (NATs) and firewalls (FWs) [Webrtc 2017].

The second one is WebTorrent protocol. WebRTC makes it possible to use advanced P2P technologies such as BitTorrent in the delivery of Web 3D contents. BitTorrent is a mesh-based protocol to P2P to transfer fine-grained data packets called chunks simultaneously. Each user of a BitTorrent mesh can download and upload chunks simultaneously to achieve the maximum utilization ratio of networking transmission under currently available bandwidth.

Since WebTorrent is based on Bitorrent with WebRTC instead of TCP/UTP, it works like BitTorrent protocol. The tracker protocol has been made a few changes to support WebRTC's connection well. Thus, a browser-based WebTorrent client or "web peer" can establish P2P connection with other Web clients to support WebTorrent/WebRTC [Feross aboukhadijeh 2017].

### *3.5.4 Workflow.*

1) All Web3d packets are seeded in advance and generate magnetic links (A magnet link contains all the information needed to download the requested files from peers directly) for WebTorrent. Considering the robustness of WebTorrent downloading, some servers are deployed at different places as an initial bunch of distributed uploading nodes, to ensure that WebTorrent can always find the downloading nodes.

2) Once a client requests for a magnetic link of some corresponding Web3d packets, its current location and the information of the required packet is sent to the server. The server will forward a query to MongoDB, then, this Web client gets the required magnetic link.

3) After getting a magnetic link, the client will call WebTorrent to parse the magnetic link and discover the neighboring peers by the aid of Bittorrent protocol.

4) From those peering candidates discovered, some suitable peers will be selected to form a multi-source P2P overlay network for Web3D transmission. Then, one peer can fetch Web3D packets from its neighbors through WebTorrent protocol.

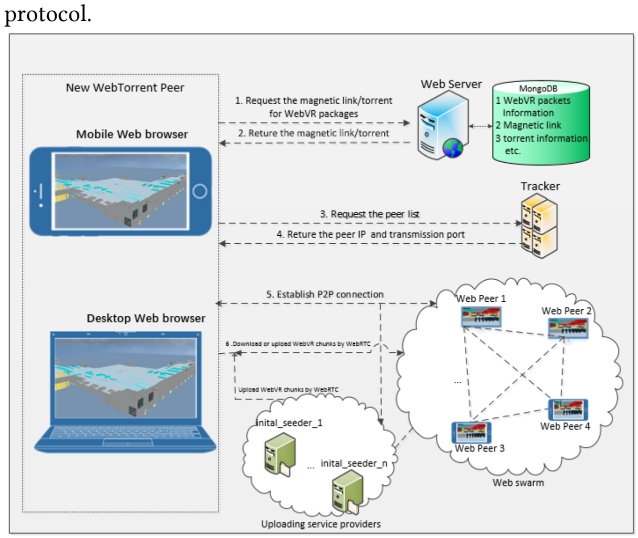



**Figure 11: Basic Sequence Diagram of Web3DTorrent**

## 3.6 Asynchronous transmission and rendering

Considering that there are too many components in each subspace usually, thereby, their loading time become long, consequently, final Web3D rendering will become rather long, a two-thread loading-rendering synchronization mechanism is proposed for real time Web3D visualization by minimizing reduction of loading latency and maximizing efficiency of Web3D rendering. The flowchart is detailed as follows.

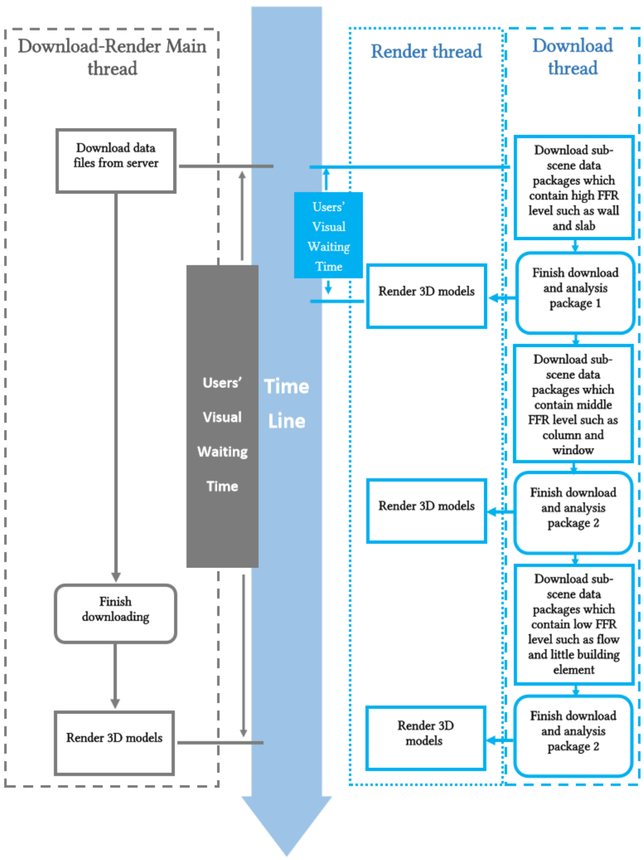



**Figure 12: Comparison of traditional loading mechanism (Left) and two-threaded synchronous loading-rendering mechanism (Right)**

In traditional serial manner of downloading-rendering in the left side, the rendering process should wait a long time for downloading all the visible Web3D models. While our synchronization mechanism in the right-side download Web3D models progressively step by step, once Web client just downloading a bit of fine-grained data successfully, it will parse and deliver these models to render immediately according to both repetitiveness and FFR. Thus, the latency of Web3D visualization is reduced effectively and thereby users feel visually much smoother.

# 4 RESULTS AND DISCUSSION

## 4.1 Dynamic Measurements: BLS

Web3D scenes can be transferred via packet as a unit with packaging mechanism. Otherwise, Web3D scenes are transferred via entity as a unit without packaging, each time a Web3D entity is requested for downloading by a Web client, it must send a downloading request to server. Comparing with the data size of transferred Web3D entity, each request contains a big header except for the requested Web3D models. Transferring this header repeatedly will waste quite much Internet bandwidth. We will show how our proposed packaging transmission reduces the number of requests from clients to server by the following example of a subspace R1-2.

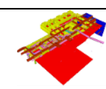
*4.1.1 Test Case.* The subspace of Chegongmiao metro subway station called R1-2 containing 1805 models is chosen to test packaging, its total size is about 6.52MB.

*4.1.2 Test configuration.* Desktop, laptop and mobile browsers are used for mobile Internet open testing. Common configuration is listed as Server OS: WinServer2008, Mem: 4G, CPU: Intel Xeon 2.39G. Client OS: Windows7, Mem: 8G, CPU: Intel i7-4700MQ. Browser: Chrome 52.0.2743. Network bandwidth: 100Mbps, campus export bandwidth of 4.5Gbps.

*4.1.3 Test steps.* 1) Test the total transferring time and data size by chrome with packaging. 2) Test the total transferring time and data size by chrome without package strategy.

*4.1.4 Test results.*

**Table 2: Comparison of packing policy**

|  | Without packaging | With packaging | Improving efficiency (Percentage) |
|---|---|---|---|
| The requests of Web3D entities | 1805 | 15 | 11933% |
| The data size of transferring Web3D entities | 8.325Mb | 5.815Mb | 43.16% |
| The total transferring time | 90~100s | 3~4s | 2400% |

Among them, the transferring data amount is calculated as follows:

$$\boldsymbol{V_n} = \sum_{i=0}^{a} (\boldsymbol{V_h} + \boldsymbol{V_e}) \qquad (2)$$

$V_n$, the total transferring data volume, is the sum of all the requested Web3D files, $V_h$ is the header size of all the transmitted Web3D files and $V_e$ is the size of all requested Web3D contents.

The actual size of a packet with 1805 files is 6.52M, the number of requests is equal to the number of Web3D models with packaging strategy and the size of the header is 1KB, So its transmission capacity should be 8.325MB = 6.52MB + 1.805MB; After using the packing strategy, the data packet is 15, the actual occupied space is 5.815MB=5.8MB+15/1000MB. Therefore, packaging decreases effectively the client requests.

## 4.2 Web3DTorrent Transmission Experiment

We combine lightweight progressive transmission and WebTorrent based P2P transmission to develop a practical Web3DTorrent platform. A large scale WebVR scene of metro station is chosen as preliminarily tested to testify to the effectiveness of Web3DTorrent. 100 users are organized to take open test of Web3DTorrent over mobile Internet.

*4.2.1 Test Case.* WebVR scene of a city metro station (The area: 1,200,000 square meters; the size: 920M). This is a huge interior space with complicated layout or structure. It is very challenging task to reach Web online downloading and real-time rendering.

*4.2.2 Test configuration.* The configuration is same with above packaging test. Server OS: WinServer2008, Mem: 4G CPU: Intel Xeon 2.39G. Client OS: Windows7, Mem: 8G, CPU: Intel i7-4700MQ. Browser: Chrome 52.0. 2743. Networking bandwidth: 100Mbps and exporting bandwidth is 4.5Gbps.

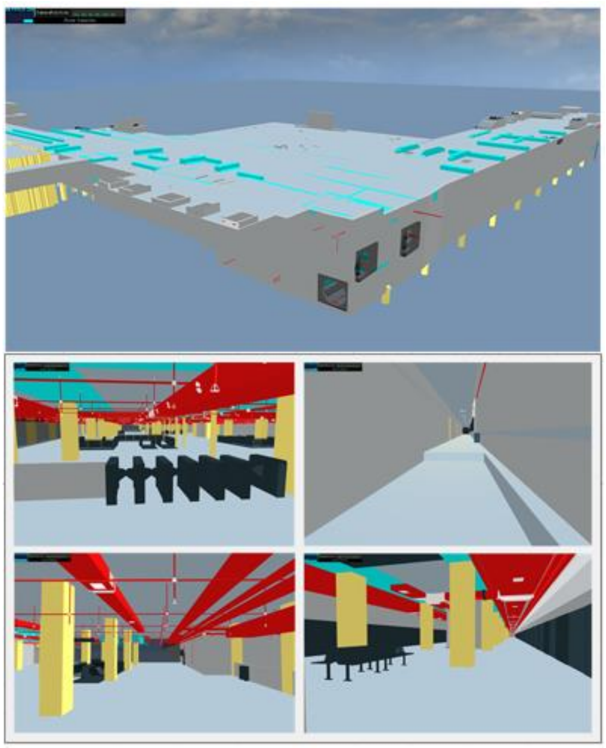


**Figure 13: A subway station outdoor overview and its four indoor screenshot**

*4.2.3 Test steps.*

1) Add the code 'console.time' to start timing at the point where the Web3D page was just loaded, and add code 'console.time' to stop timing at the point where the rendering is complete. 2) Enter the URL: http://smart3d.tongji.edu.cn/TongyanTest/index.html 3) View the loading time displayed in the console.

*4.2.4 Experimental results.*

Unlike majority of existing transmission experiments of DVE-P2P were done onto simulation platform not real tests over Internet or mobile Internet, we conducted a real public test onto mobile Internet and obtained some convincing experimental results as follows. We did our best to organize a team of 100 people to access Metro Station scenario simultaneously.

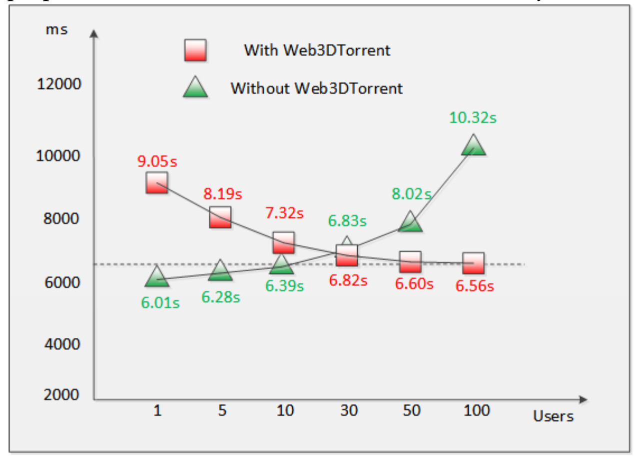



**Figure 14: Comparison of downloading time tested with Web3DTorrent and without Web3DTorrent**

*4.2.5 Performance analysis.*

From Figure 14: Comparison of downloading time tested with Web3DTorrent and without Web3DTorrent, we can see that non-Web3DTorrent's downloading time varies with the number of users, Web3D transmission performance is decreasing obviously with the number of users growing, once 100 users accessing the same metro station simultaneously, the server crashes and the Internet congestion happens.

On the other hand, when Web3DTorrent is deployed, the downloading time is slightly higher than non-Web3DTorrent due to the time overhead of initial Web3DTorrent loading. However, with the number of users increasing, Web3DTorrent transmission strategy advantages over non-Web3DTorrent obviously. Also, it reduces greatly the pressure of accessing servers, even when server crashes until 100 users online accessing simultaneously, users can still access Web3D metro station by the aid of Web3DTorrent.

## 5 CONCLUSIONS AND FUTURE WORK

This paper establishes a novel Lightweight P2P transmission of large-scale WebVR scenes by the aid of mature WebTorrent. We have successfully implemented large-scale WebVR scenes over mobile Internet. Practical experiments show that the proposed scheme obviously improves WebVR transmission performance over mobile Internet although our initial Web3DTorrent framework is not a complete or ripe work. We will gradually improve it step by step in the future in terms of the following points.

### 5.1 Automatic fine-grained preprocessing

Now the fine-grained preprocessing is taken artificially according to FFR level to determine which areas are closed or not, the efficiency goes relatively low, so we will continue to study an automatic preprocessing of splitting a large VR scene into smaller subspaces, according to the adjacency, closeness and lightweight.

## 5.2 Adaptive Packaging with Web-Pack

The current mechanism is based on FFR and has not taken into account networking bandwidth, so we will use advanced Web-Pack to classify existing FFR according to transmission bandwidth of server and adaptively pack. Such a dynamic WebVR packaging is more reasonable.

## 5.3 Building a Tracker Server

Now, our Web3DTorrent platform employed the foreign Tracker server to manage indexing and retrieving of peers. The efficiency is rather low due to long time to transfer. Next stage, we will build our own tracker server locally for more efficient management of peer indexing and retrieving.

## ACKNOWLEDGMENTS

The authors appreciate the comments and suggestions of all anonymous reviewers, whose comments significantly improved this paper.

This work is supported by The Key Research Projects of Central University of Basic Scientific Research Funds for Cross Cooperation (201510-02), Research Fund for the Doctoral Program of Higher Education of China (No.2013007211-0035) and Key project in scientific and technological of Jilin Province in China (No.20140204088GX).

## REFERENCES

Xml3d http://xml3d.org/
X3dom https://www.x3dom.org/
Cobweb http://titania.create3000.de/cobweb/
Three.js htpp://threejs.org
Gltf https://github.com/khronosgroup/gltf/
A-frame vr https://aframe.io/
Shikhare d, bhakar s, mudur s p. Compression of large 3d engineering models using automatic discovery of repeating geometric features[j] signal processing, 2001, 19( 20) : 233-240.
Laixiang Wen, Ning Xie, Jinyuan Jia, "Client-Driven Strategy Of Large-Scale Scene Streaming", Multimedia Modeling (Mmm), 2015
Michael englert, yvvone jung, paul grimm and jinyuan jia. "a streaming framework for instant 3d rendering and interaction". Acm vrst. Beijing, china. November 2015.
Xiaojun liu, nine xie, kai tang, jinyuan jia. "lightweight for web3d visualization of large-scale bim scenes in real-time", graphical models, vol. 88, 2016.
Buyukkaya, e., et al. A survey of peer-to-peer overlay approaches for networked virtual environments. Peer-to-peer networking and applications[j]. Pp: 1-25. 2013.
Yahyavi, a. And b. Kemme. Peer-to-peer architectures for massively multiplayer online games: a survey. Acm computing surveys 1: 8022980[j]. 2013, 46(1):1-45.
Hu s y, huang t h, chang s c, etc. "flod: a framework for peer-to-peer 3d streaming". Infocom'08. Phoenix, usa, 2008
Jia jy, wang w, hei x. An efficient caching algorithm for peer-to-peer 3d streaming in distributed virtual environments[j]. Journal of network & computer applications, 2014, 42(3):1–11.
Mingfei wang, jinyuan jia, yunxiao zhongchu and chenxi zhang, "interest-driven avatar neighbors organizing for p2p transmission of distributed virtual world". Computer animation and virtual worlds. (doi:10.1002/cav.1670). 2015.
Webrtc http://www.webrtc.org
Feross aboukhadijeh. 2017. What is webtorrent. Https://webtorrent.io/
Lopez fernandez l, paris diaz m, benitez mejias r, et al. Kurento. A media server technology for convergent www/mobile real-time multimedia communications supporting webrtc[c]. The 14th ieee symposium on world of wireless, mobile and multimedia networks (wowmom), pp. 1-6. 2013.
Lopez fernandez l, diaz m p, benitez mejias r, et al. Catalysing the success of webrtc for the provision of advanced multimedia real-time communication services[c] (icin). The 17th ieee intelligence in next generation networks. Pp. 23-30. 2013
Kiran kumar guduru. "webrtc implementation analysis and impact of bundle feature". The fifth ieee conference on communication systems and network technologies 2015
Haroula andrioti, andreas stamoulias and kostas kapetanakis. "integrating webrtc and x3dom: bridging the gap between communications and graphics". Acm web3d 2015. Pp. 9-15, crete island, greece. June 2015.