Towards Web3D-based Lightweight Crowd Evacuation Simulation

Xue Piao
School of Software Engineering, Tongji University
Shanghai, China
piaosally@gmail.com

Yang Li
School of Software Engineering, Tongji University
Shanghai, China
qw876056236@gmail.com

Kang Xie
School of Software Engineering, Tongji University
Shanghai, China
xkdemail@gmail.com

Hantao Zhao
School of Cyber Science and Engineering, Southeast University
Nanjing, China
Chair of Cognitive Science, ETH Zurich
Zurich, Switzerland
hantao.zhao@gmail.com

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

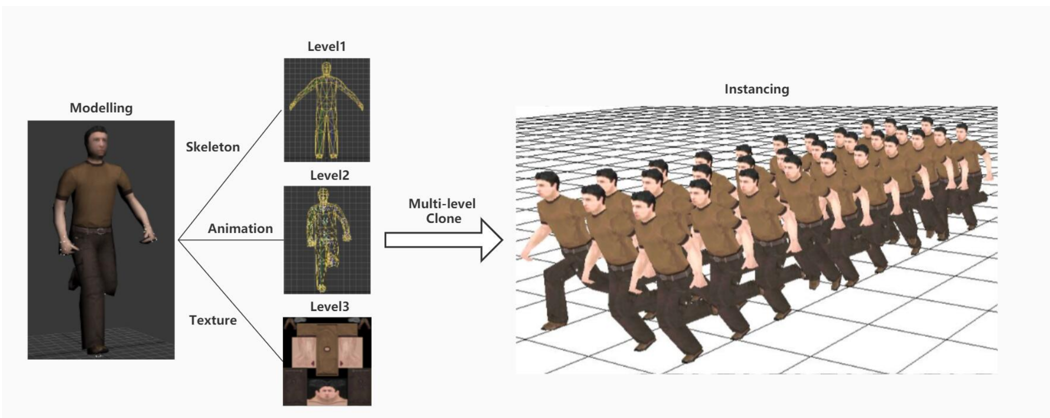

Figure 1: Multi-level clone instancing technique

ABSTRACT

The heterogeneity of the appearance and behavior of the crowd is an important component in a realistic simulation. Using Web3D technology to recreate a large-scale crowd of virtual avatars via the internet is an increasingly important area in crowd studies, especially for emergency crowd evacuation. One major issue in the early development of real-time simulation is the challenge of introducing diversity and authenticity on avatars' appearance and behavior. Another major theoretical issue is high transmission delay on the network, which diminishes the simulation performed on the web browser when rendering a large number of visualization components. In this research, we propose a novel method to provide a lightweight solution that can simultaneously guarantee the realness and diversity of the simulated crowd. We use a parameterization technique based on shape space to distinguish the avatars' appearance. Asynchronously transmitting the rendering elements helps to reduce bandwidth pressure. The multi-level clone instancing method can generate a massive amount of heterogeneous avatars in a short time. In the end, we validated our methods with an online experiment to demonstrate its ability to solve large-scale crowd simulation problems over the internet.

CCS CONCEPTS

• **Computing methodologies → Simulation environments**.

KEYWORDS

web3D, web browser, crowd simulation, evacuation behavior, model parameterization, avatar rendering

Web3D '20, November 9–13, 2020, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8169-7/20/11...$15.00
https://doi.org/10.1145/3424616.3424708

ACM Reference Format:
Xue Piao, Yang Li, Kang Xie, Hantao Zhao, and Jinyuan Jia. 2020. Towards Web3D-based Lightweight Crowd Evacuation Simulation. In *The 25th International Conference on 3D Web Technology (Web3D '20), November 9–13, 2020, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3424616.3424708

1 INTRODUCTION

Rendering large-scale crowded scenes via a web browser has become a commonplace due to the convenience of its lightweight solution. With help from the ubiquitous existence of personal electronic devices, it becomes common to access the simulated scenes through a web browser instead of downloading and installing an independent executable or plugin. However, due to the inherent shortage of computing resources from in browser's end, the difficulty of loading and rendering the behaviors of large-scale crowds in real-time can not be neglected. Traditionally, the avatars of the characters from the modeling software can not be altered freely, resulting in a lack of diversity among them. A group of homogeneous avatars do not reflect the realistic dynamics of real crowds. Therefore, it is worth investigating novel methods and techniques in the real-time simulation of large-scale crowd behavior in Web3D.

In this paper, a model parameterization technique based on shape space was combined with a multi-level clone instantiation technique. The proposed solution tackles the issue of loading and rendering a vast and differentiated dynamic population with limited resources on the browser side. This method not only doubles the maximum number of rendered avatars that the traditional way can achieve on the browser but also significantly reduces the amount of memory resource used by the same amount of workload. We have found that it significantly reduces the initial load time and increases the rendering frame rate. The parameterization also increases the diversity of the avatars in terms of bone shape, action, and texture data. This lightweight and diverse approach establishes future standards for Web3D crowd simulation tools.

2 RELATED WORK

Nowadays, crowd visualization technologies are commonly agent-based through simulations [Oğuz et al. 2010; Pluchino et al. 2013]. The disadvantage of using a separate model is that the loading time can be very long, causing the frame rate to drop once the number of avatars increases. By far, the previous approaches focus on simplifying the model itself or improving the facility of the hardware. Often this kind of approach neglects the diversity and authenticity of large-scale crowd simulation. The concept of model parameterization at present has been applied to the simplification of buildings in the Web3D Building information modeling (BIM) [Liu et al. 2016; Yan et al. 2019]. In addition, the behavior of evacuation were investigated by researchers [Zhao et al. 2020; Zheng et al. 2013; Zhu and Shi 2016]. As the dynamic characters appear more and more often in the internet space, the parameterization of the dynamic crowd in large-scale scenes requires more attention.

The solution to the problem of large-scale scene transmission and rendering is always a challenge. There exists research on web3D large-scale dynamic scene lightening [Kim and Park 2015; Peeta et al. 2011; Tian et al. 2014; Wen et al. 2014]. In recent years, network transmission, graphics-based progressive transmission and networking based peer-to-peer transmission [Hu et al. 2017a] have improved the speed of networking communication. The appearance of multi-core processors and many-core processors and the positive development of GPU (Graphics Processing Unit) technology provides a feasible solution for the simulation of large-scale group movement. Meanwhile, parallel computing has been increasingly applied in crowd motion simulations [Zhao et al. 2019]. Malinowski et al. proposed to a parallel environment to simulate and visualize evacuation scenarios [Malinowski et al. 2017], and other researchers used the concept of modular design [Malinowski and Czarnul 2019]. With the assistance of the GPU, the actual running speed of the environment is faster[Haciomeroglu et al. 2013]. GPU instance mechanism and LOD (Levels of Detail) technology were used to achieve the real-time rendering of thousands of avatars, but the simulated avatars were homogeneous [Savoy et al. 2015]. The diversity and reality factors were not taken into account in most of the previous studies. It becomes necessary to establish a method to flexibly enhance the efficiency with the further expansion of crowd size through Web3D, without violating the requirements of diversity and authenticity.

3D human modeling is a well researched domain. The appearance of 3D scanning technology promotes the rapid development of human parametric modeling. Anguelov et al. proposed Shape Completion and Animation for People (SCAPE) to tackle this challenge [Anguelov et al. 2005]. Since then, there have been studies on optimization based on the SCAPE method [Chen et al. 2015; Yang et al. 2016], and dynamic capture has also been used in research in this field [Bogo et al. 2016; von Marcard et al. 2017; Zhang et al. 2017]. In addition, deep learning methods [Dibra et al. 2016a, 2017, 2016b; Ghezelghieh et al. 2016] have also been applied in 3D human modeling and demonstrated satisfactory results. Bermano et al. proposed a projector-based lighting processing system[Bermano et al. 2013]. Weise et al. proposed a novel facial tracking algorithm to pursue the authenticity and expressiveness of the 3D avatar's movements and expressions[Weise et al. 2011]. Hu et al. used deep convolutional neural network technology to improve the generated 3D avatar effect[Hu et al. 2017b]. Nagono et al. generated a high-quality 3D avatar appearance based on a new Generative Adversarial Network (GAN)[Nagano et al. 2018]. These three-dimensional shape parameterization methods were used to represent different body shapes to replay different postures or build new shapes that are not available in the training data set [Allen et al. 2003; Cheng et al. 2018]. Ai and colleagues have forwarded the idea of crowd parameterization, but the preliminary approach brought disadvantages in diversity and scalability [AI et al. 2019] . Therefore, it was apparent that the diversity of appearances and movements of virtual avatars were need of improvement. Moreover, the model from Ai and colleagues depends on heavy computation, causing the number of properly rendered avatars was too low. The maximum number of rendered avatars was below one thousand, which could not meet the need for a large-scale group's real-time simulation.In crowd simulation and rendering realm, Maim and colleagues developed an application using efficient algorithm and technique based on segmentation maps to generate unique avatars[Maim et al. 2009; Maïm et al. 2009]. While aiming at real-time simulation on the web, we used parameterization and cloning technology to generate thousands of varied virtual humans navigating in large-scale scenario online.

3 METHOD OVERVIEW

Together, these studies indicate that research to date has not yet tackled the issues of rendering a large crowd through Web3D technology. This paper addresses this challenge through the following three different perspectives.

Diversity and reality of large-scale crowd. Traditional methods used already existed models exported from professional software. The realness of a massive group comes from a large number of models with different appearances and actions, which results in a large amount of loading computation during modeling. Moreover, limited models can hardly meet the diverse requirements. To solve this bottleneck, we develop a parameterization technique based on shape space. It parameterizes the bone shapes, texture, and animations. The parameters can be adjusted within a range at will according to the actual situation. Therefore, we can generate virtual avatars with various personalities using just a few original models.

High network transmission delay. Large-scale data uploading and downloading through the web brings a significant delay due to the limited bandwidth and the computational time of simulations. The waiting time for clients who visit the website would be ideally within three seconds. Once the delay exceeds this amount of time, the client's experience would be significantly affected. To address this problem, we used a progressive loading algorithm and asynchronous loading to relieve the bandwidth pressure.

Rendering performance. Dynamic scenes with animated avatars require large data volume and can cause computational pressure on the web browser. A set of lightweight solutions is needed in online real-time rendering. Based on a lightweight scheme, combined with a multi-level clone parameterization technique, a small number of original models can be used to generate large quantities of avatars by cloning. Instead of loading fixed models one by one like conventional methods, clone instancing shortens the time and guarantees the smoothness of rendering enormously.

4 LIGHTWEIGHT WEB3D SIMULATION FRAMEWORK

A lightweight Web3D simulation framework is proposed here to improve the performance of the web crowd rendering, and to fulfill the above standards. It combines the above advantages with the multi-level clone instanced rendering. In this implementation, we present a server for single mode structure analysis. From the browser end, a multi-level clone instancing method is used to generate large-scale models. At the same time, random parameterization of the structural nodes of each model is applied to generate new models with different appearances. Furthermore, we attach different textures and change animations to make the motion of each model diverse. The detailed workflow is shown in Figure 2 below. Figure 3 is the EC-GLB (Extended Character-GLB) model node tree proposed according to the structural characteristics of the character model in GLB format. This node tree is also the key to the subsequent implementation of the parameterization algorithm. The detailed parameterization methods can be divided into the following five steps:

- Modeling – Use a blender to model and name each level of nodes, so that the hierarchy is ordered and managed in a controllable way.
- Generate the model node tree – In the exported GLB model, the actual model joint is positioned according to the nodes in the node tree. In Figure 2, the node tree of the GLB model is drawn, and the nodes related to corresponding parameterization technology are indicated.
- Structure analysis from the front end – After the front end accepts the model, it traverses the structure tree and finds the node information (e.g., arm, leg, etc.) that can be parameterized.
- Parameterized model generation – After getting the node information of the model, the front end uses the clone technique to instantiate the model on a large scale.
- Model diversity rendering – Set the parameters of the animation in the animation library and bind the designated animations to the model.

4.1 Data transmission

Traditional 3D model files (i.e., Obj, FBX, COLLADA, etc.) are suboptimal for transmission over the internet due to their nature of large volume. In addition, a complex parsing operation is needed to render the web page. In this data transmission optimization approach, we choose to use the glTF(GL TransmissionFormat) format as a model format. It was developed and maintained by the Khronos organization with the compatibility of being lightweight. glTF is essentially a file in JSON(JavaScript Object Notation) format, which is very convenient for the browser to parse. It abstracts the complete 3D scene into a tree structure based on hierarchical nodes. Figure 4 shows the data organization structure in the glTF format. The redundant fields which are irrelevant to rendering are eliminated. The geometry, animation, and mapping information of the model are stored in a binary mode that can be computed by GPU. In this way, the glTF format file not only has a small amount of data but also greatly improves the transmission and loading speed.

Furthermore, glTF also provides the convenience of being a binary form, which can store data into separate files. Binary format takes less space than text format, which further reduces the storage occupied by the model. The dynamic character models used in our project are low-poly 3D models in GLB format.

4.2 Multi-level clone instancing

Further cloning techniques were implemented to help to generate and multiply the avatars until the targeted number is reached. Because Three.js has no clone method for GLB format, it needs to be implemented independently. We create a new object and then clone the internal parts in turns. Algorithm 1 shows the implementation of the cloning mechanism. The parts that need to be cloned contains the animations, bone, and skinned mesh. The cloning of animation is different from the bone and skinned mesh, which need a separate cloning. It also demonstrates the specific differences in the process.

Figure 1 demonstrates an example of GLB human model cloned into several identical models. The skeleton and animation of each model are independent and do not affect each other. The animation playback speed is randomly set. It can be seen in the figure that the avatars' arm swing ranges and step sizes have heterogeneous characteristics.

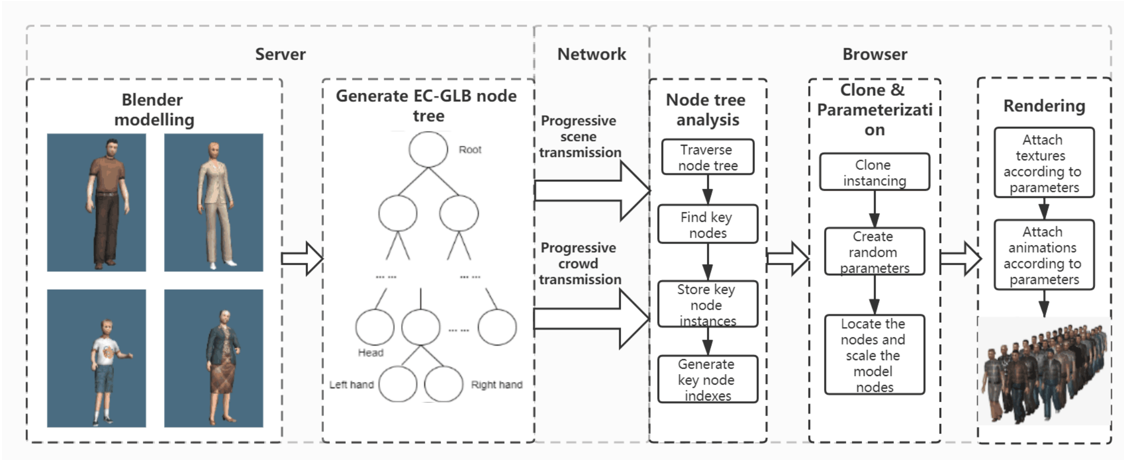

Figure 2: Workflow for model parameterization

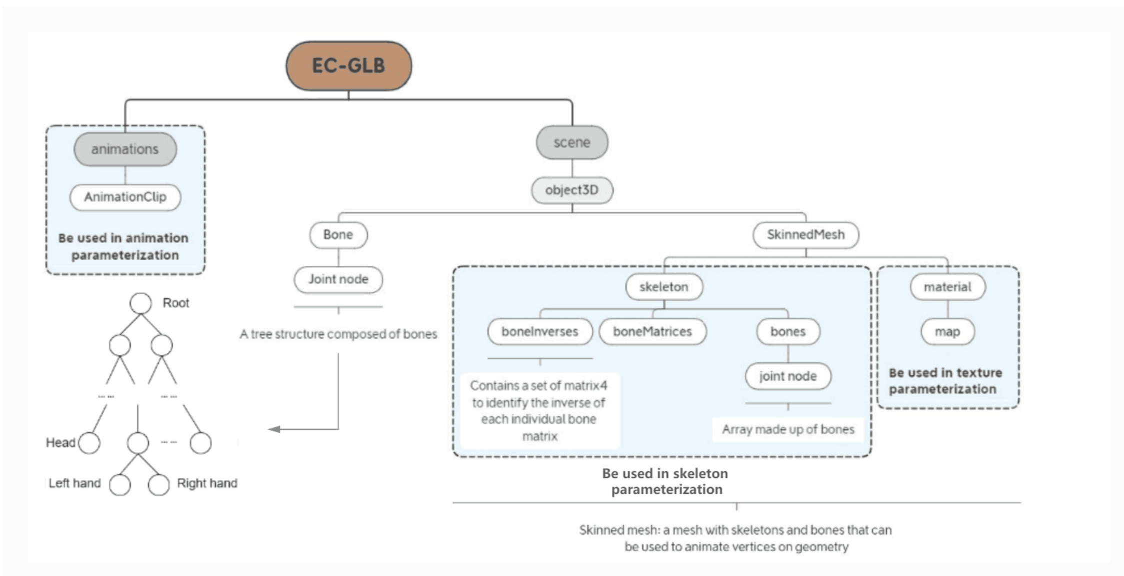

Figure 3: Technical roadmap

4.3 Model parameterization

Last but not least, we introduce the parameterization of texture, skeleton, and animation. It relies on the original model but can reduce the usage of bandwidth and loading time by using an asynchronous computation and loading function. The following algorithm 2 describes the process of a character parameterization. Different algorithms of texture parameterization, skeleton parameterization, and animation parameterization are applied, respectively.

For the parameterization of the GLB model texture, body shape, and animation data, we implemented a model parameterization generator in the framework. As Figure 5 shows, the generator converts a series of parameters into diverse character models through a module.

There exist three parameter generators in the module, which are texture generator, skeleton scale generator, and animation data generator. The input "newMesh" parameter stores the original model that is not parameterized, and "temp" is the texture serial number of the current model, which determines what texture should be selected for replacement in the texture generator. The model is an input to the bone scale generator after replacing the texture. Here, the EC-GLB node tree proposed in Figure 3 is used to locate the skeleton to be parameterized in the structure tree joints (e.g., head, neck, shoulder, abdomen, etc). The random function can further parameterize the node. The specific parameters are adjusted according to ergonomics and visual effects. When all the bone nodes are adjusted, the model is input to the animation data generator. First, it extracts the unique animation of the model by changing some

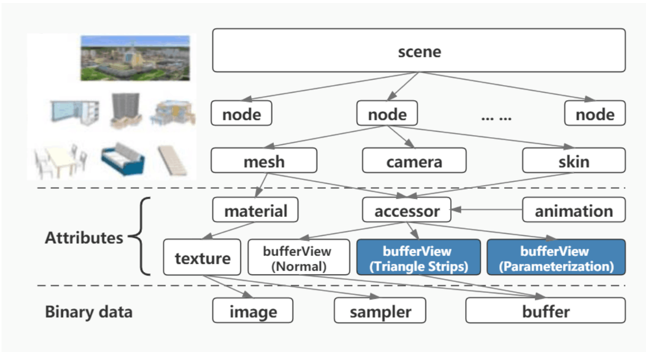

Figure 4: Data organization structure of glTF format

Algorithm 1: Bones and Skinned Mesh Cloning Algorithm

```
Input:gltf;
clone = {
        animations : gltf.animations
        scene : gltf.scene.clone(true)
};
skinnedMeshes={};
for each node of gltf.scene do
    if ForceGetProperty( node, 'isSkinnedMesh' ) then
        skinnedMeshes[ node. uuid ] = node;
    end
end
for each node of clone.scene do
    if ForceGetProperty( node, 'isBone' ) then
        cloneBones[ node.uuid ] = node;
    end
    if ForceGetProperty( node, 'isSkinnedMesh' ) then
        cloneSkinnedMeshes[ node.uuid ] = node;
    end
end
for each uuid of cloneSkinnedMeshes do
    cloneSkinnedMesh = cloneSkinnedMeshes[ uuid ];
     skinnedMesh = skinnedMeshes[
     cloneSkinnedMesh_sourceMeshUuid];
    if skinnedMesh = null then
        continue;
    end
    skeleton = skinnedMesh. skeleton; for each bone of
     skeleton do
        cloneBone = cloneBones[ skeleton. bone. name];
         orderedCloneBones. push( cloneBone );
    end
    cloneSkinnedMesh.bind( new Skeleton(
     orderedCloneBones, skeleton. boneInverses
     ),cloneSkinnedMesh.matrixWorld );
end
Output:clone;
```

characteristics of the animation key frame. The time and values parameters of the particular frame can be altered to generate a new

Algorithm 2: Shape Space Based Model Parameterization Algorithm

```
Input:modelURL,n,sum;
i=0;
for i<n do
    arr[i] = loadModelPromise(modelURL);
    i++;
end
promiseAll = Promise.all(arr).then((data)=>
i=0;
for i<sum do
    temp = i%n;
    newMesh = cloneGltf(data[temp]);
    Perform parameterization module;
    scene.add (newMesh.scene);
    i++;
end
);
Onput:Various parameterization GLB models;
```

animation based on its original one, but with randomized animation playback speed. In the end, the animation is activated and repeated automatically. The "newMesh" has been fully parameterized and a new model different from the original model has been generated. In Algorithm 2, the basic model is instantiated and cloned, and then the "module" is repeatedly called to achieve the generation of the large-scale diverse population.

The combined approaches successfully achieved the requirements of lightweight from three levels: the server reduces the amount of data by model pre-processing; the network alleviates the bandwidth consumption through the progressive asynchronous loading; and the web cuts down the calculation through the clone instancing. By just loading sample models, we can generate a large number of distinct avatars through the web. This model pasteurization greatly reduces the initial loading time and improves the rate of rendering frames. Lastly, the model parameterization technology based on shape space enables the diversity and reality of large-scale crowd behavior simulation by introducing varieties of texture, skeleton, and animation.

5 RESULTS

In order to examine and validate the framework, we deployed the framework into a real-world scenario of a large-scale virtual avatars simulation. We implemented the whole framework as an online demo (visit https://smart3d.tongji.edu.cn/crowdDemo/fire/examples/fire.html) and validated it through a series of the experiment in comparison with prior work.

The results of the texture parameterization and the skeleton parameterization of the avatars are respectively shown in Figure 6 and Figure 7. Figure 6 shows the course of the variation of a crowd of avatars from single texture to a variety of new textures through diversification. In figure 7, it can be seen that the sizes of the head and abdomen are different from the original models. Similarly, other parts of the body are changed for full skeleton

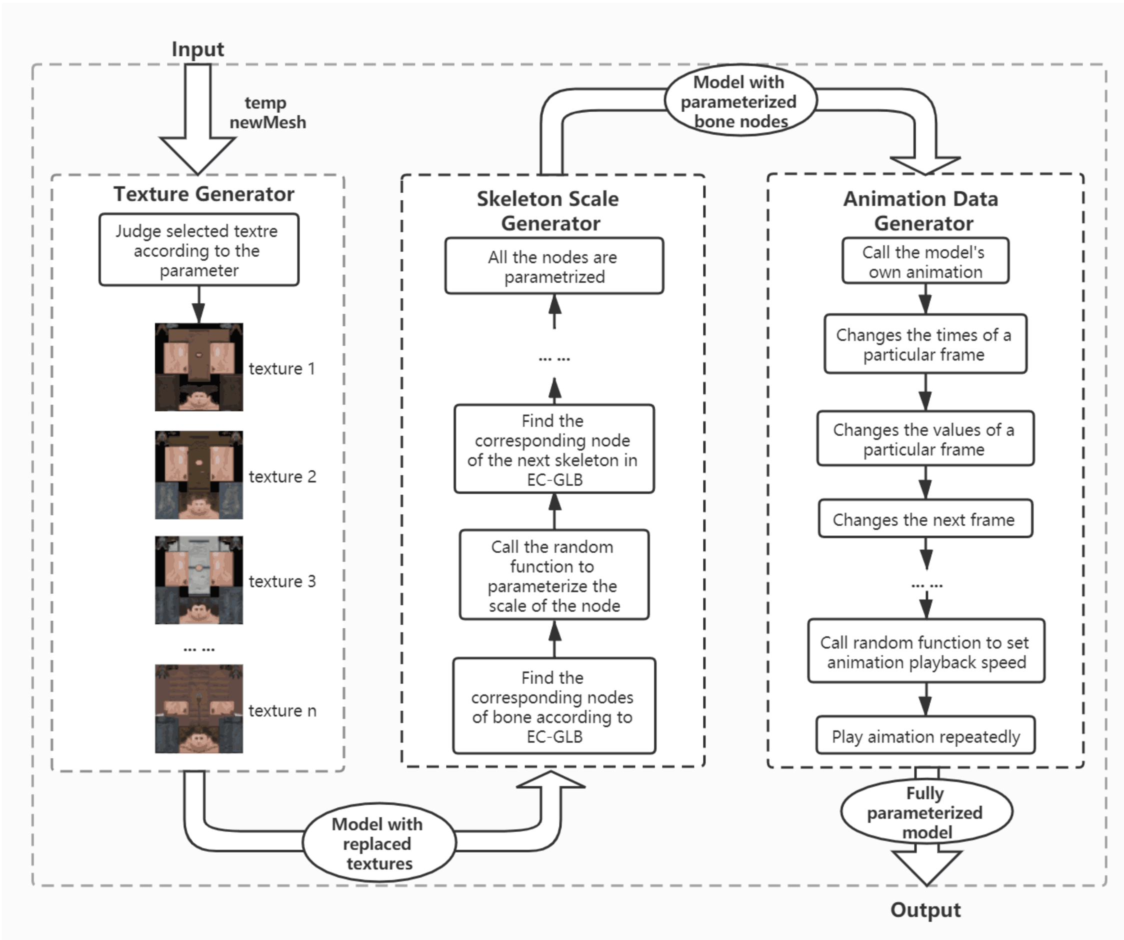

Figure 5: The internal logic of the module

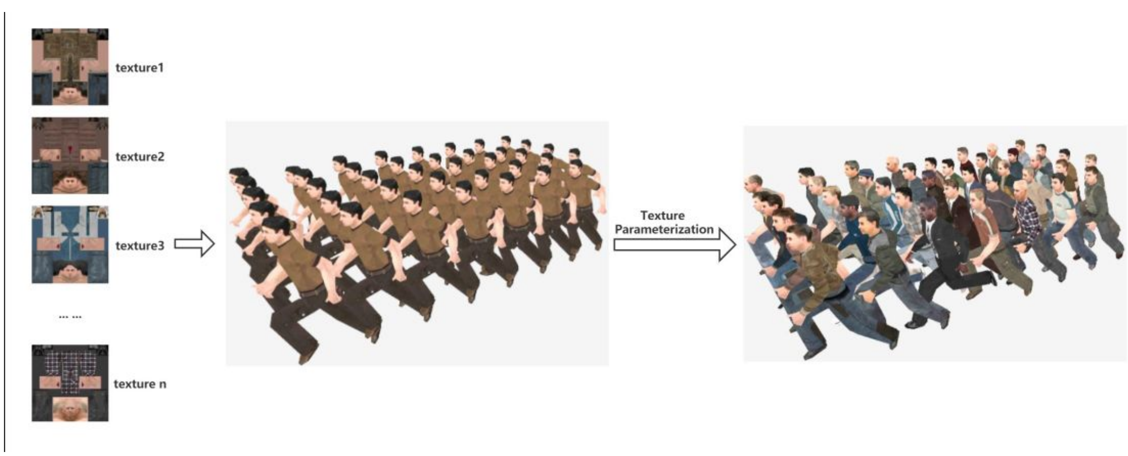

Figure 6: Texture parameterization: from single texture to a range of textures

parameterization. It also shows that the running, walking, bending, and crawling actions are formed after parameterization from initial walking action. The change of a group of avatars with a range of new textures after the texture parameterization can be seen Figure

8 shows the real-time rendering of thousands of diversified avatars in a simulation scene of fire evacuation.

In order to demonstrate the performance improvement, we compared our approach with the previous method [AI et al. 2019] via different computing platforms. The test platforms are listed in Table 5. Three measures were used for an overall evaluation:

(1) The initial loading time of model;
(2) Memory consumption during scene loading;
(3) The rendering frames rate during scene loading.

Table 1: Hardware configuration in test environment.

	PC Testing Environment	iOS Testing Environment	Android Testing Environment
CPU:	i7-7700HQ	Apple A10	Snapdragon 820 (MSM8996)
Memory:	16GB	2GB	4GB
GPU:	Nvidia GTX M1070	PowerVR GT7600	Adreno530
OS:	Windows 10 64Bit	iOS 11.4	Android 6.0
Networks:	4G Wireless Network	4G Wireless Network	4G Wireless Network
Browser:	Chrome	Chrome	Chrome

The dotted line is the testing results measured with the prior method [AI et al. 2019], and the solid line is the testing results with the present method. As demonstrated in Figure 9, the previous method can achieve a peak number of 500 avatars, whereas our method extends the peak performance to 3000 avatars. With a comparison of two sets of data, it can be seen that the present model is more lightweight. Rendering the same number of avatars requires fewer memory resource. In addition, the rendering frame rate of our method with 500 avatars is much higher than that of the prior method, without diminishing the rendering performance. Moreover, it shows that when loading up to 3000 avatars, the waiting time of web browser is less than 3 seconds, which guarantees a satisfactory user experience.

Overall, the present method outperforms the prior approach. First, it can vastly increase the peak number of clients that can be reached in traditional methods. Second, it significantly reduces the memory resource occupied per model and the initial loading time. Third, it greatly improves the rendering frame rate performance. Last but not least, the proposed methods achieved all of the above goals by introducing a great deal of diversity and authenticity in terms of avatar appearances, shapes, actions, and textures. The framework can indeed fulfill the requirements of authenticity for large-scale crowd behavior simulation through Web3D.

6 CONCLUSIONS

In the present study, we proposed a crowd simulation solution that combines dynamic model parameterization with cloning to address the challenge of rendering large-scale and diversified animation models using limited resource in the web browsers. The proposed framework meets the needs of lightweight, diversification, and authenticity. To our knowledge, it is the first approach to achieve the GLB cloning method and human model parameterization through Web3D technologies. The data from testing experiments further support the conclusion that the present method outperformed the prior method. It maximizes the number of avatars that Web3D can render and increase the heterogeneity of the crowd.

This study was limited by the absence of other novel methods towards Web3D-based lightweight crowd evacuation. Due to this reason, it was impossible to compare our approach with any existed methods or data sample. In future works, we will further explore the potential of the proposed framework, and provide more valuable experimental data in this area. It can be useful to implement new techniques to adapt the rendering strategy in a more intelligent way to adapt the browser. The frustum culling algorithm can help to render only the objects inside of the current range. Furthermore, this method can also be extended to simulate more complex crowd's behavior, for example personal activities, social activities and dynamic interactions between the crowds. We will apply the technology to diverse scenarios to verify its compatibility in the future work.

REFERENCES

Zihao AI, Yonghao HU, Fengting YAN, Huijuan ZHANG, Dongqing WANG, Shenglan QING, Hehua ZHU, and Jinyuan JIA. 2019. Key technology of lightweight Web3D online planning of metro fire escape. *SCIENTIA SINICA Informationis* 49, 4 (2019), 405–421.

Brett Allen, Brian Curless, and Zoran Popović. 2003. The space of human body shapes: reconstruction and parameterization from range scans. *ACM transactions on graphics (TOG)* 22, 3 (2003), 587–594.

Dragomir Anguelov, Praveen Srinivasan, Daphne Koller, Sebastian Thrun, Jim Rodgers, and James Davis. 2005. SCAPE: shape completion and animation of people. In *ACM SIGGRAPH 2005 Papers*. 408–416.

Amit Bermano, Philipp Brüschweiler, Anselm Grundhöfer, Daisuke Iwai, Bernd Bickel, and Markus Gross. 2013. Augmenting physical avatars using projector-based illumination. *ACM Transactions on Graphics (TOG)* 32, 6 (2013), 1–10.

Federica Bogo, Angjoo Kanazawa, Christoph Lassner, Peter Gehler, Javier Romero, and Michael J Black. 2016. Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In *European Conference on Computer Vision*. Springer, 561–578.

Yin Chen, Zhi-Quan Cheng, Chao Lai, Ralph R Martin, and Gang Dang. 2015. Realtime reconstruction of an animating human body from a single depth camera. *IEEE transactions on visualization and computer graphics* 22, 8 (2015), 2000–2011.

Zhi-Quan Cheng, Yin Chen, Ralph R Martin, Tong Wu, and Zhan Song. 2018. Parametric modeling of 3D human body shape—A survey. *Computers & Graphics* 71 (2018), 88–100.

Endri Dibra, Himanshu Jain, Cengiz Öztireli, Remo Ziegler, and Markus Gross. 2016a. Hs-nets: Estimating human body shape from silhouettes with convolutional neural networks. In *2016 fourth international conference on 3D vision (3DV)*. IEEE, 108–117.

Endri Dibra, Himanshu Jain, Cengiz Oztireli, Remo Ziegler, and Markus Gross. 2017. Human shape from silhouettes using generative hks descriptors and cross-modal neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4826–4836.

Endri Dibra, Cengiz Öztireli, Remo Ziegler, and Markus Gross. 2016b. Shape from selfies: Human body shape estimation using cca regression forests. In *European conference on computer vision*. Springer, 88–104.

Mona Fathollahi Ghezelghieh, Rangachar Kasturi, and Sudeep Sarkar. 2016. Learning camera viewpoint using CNN to improve 3D body pose estimation. In *2016 fourth international conference on 3D vision (3DV)*. IEEE, 685–693.

Murat Haciomeroglu, Oner Barut, Cumhur Y Ozcan, and Hayri Sever. 2013. A GPU-assisted hybrid model for real-time crowd simulations. *Computers & graphics* 37, 7 (2013), 862–872.

Liwen Hu, Shunsuke Saito, Lingyu Wei, Koki Nagano, Jaewoo Seo, Jens Fursund, Iman Sadeghi, Carrie Sun, Yen-Chun Chen, and Hao Li. 2017b. Avatar digitization from a single image for real-time rendering. *ACM Transactions on Graphics (ToG)* 36, 6 (2017), 1–14.

Yonghao Hu, Zhaohui Chen, Xiaojun Liu, Fei Huang, and Jinyuan Jia. 2017a. WebTorrent based fine-grained P2P transmission of large-scale WebVR indoor scenes. In *Proceedings of the 22nd International Conference on 3D Web Technology*. 1–8.

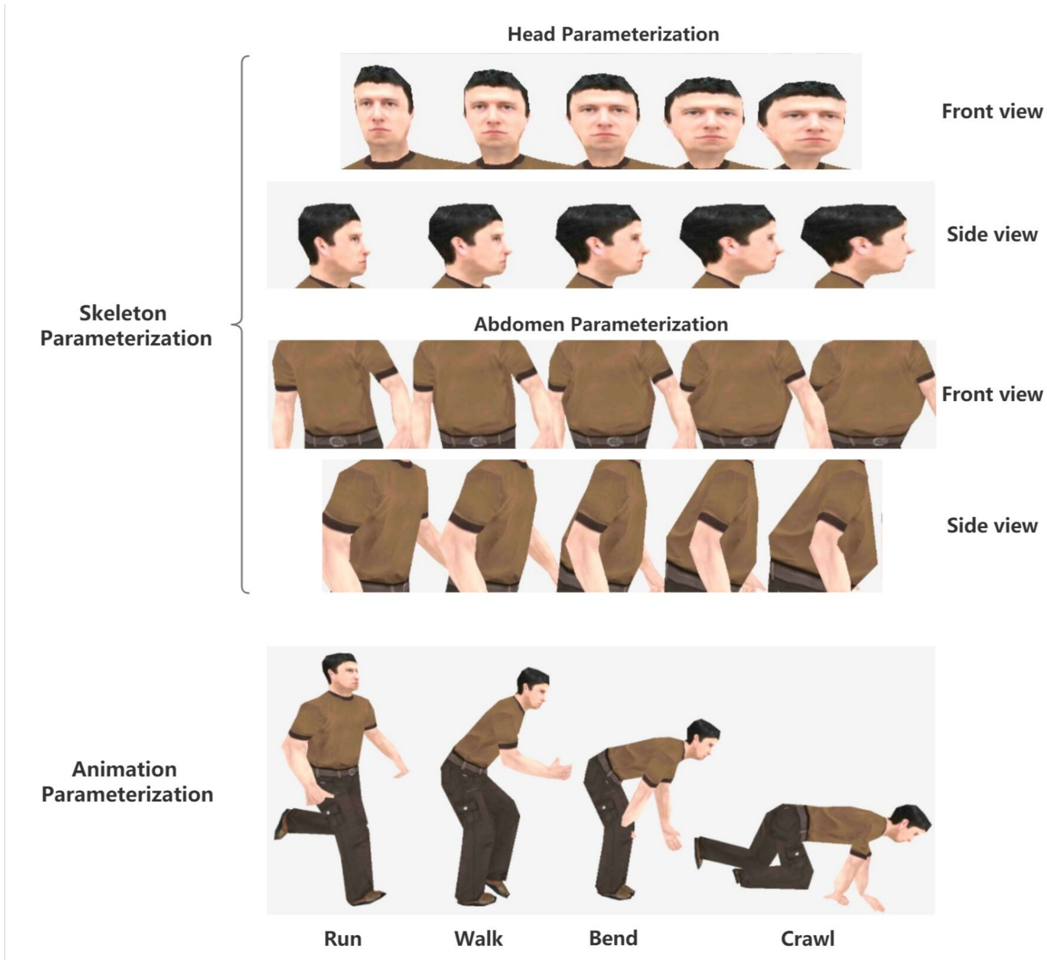

Figure 7: Animation parameterization: Run->Walk->Bend->Crawl


(a) run


(b) bend

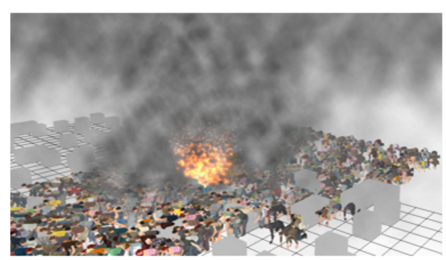
(c) crawl

Figure 8: Large scale crowd escape simulation scene

Won Ouk Kim and Woe-Chul Park. 2015. A study on grid aspect ratio of fire dynamics simulator. *Journal of the Korean Society of Marine Engineering* 39, 9 (2015), 923–928.

Xiaojun Liu, Ning Xie, Kai Tang, and Jinyuan Jia. 2016. Lightweighting for Web3D visualization of large-scale BIM scenes in real-time. *graphical models* 88 (2016), 40–56.

Jonathan Maim, Barbara Yersin, Julien Pettre, and Daniel Thalmann. 2009. YaQ: an architecture for real-time navigation and rendering of varied crowds. *IEEE Computer Graphics and Applications* 29, 4 (2009), 44–53.

Jonathan Maïm, Barbara Yersin, and Daniel Thalmann. 2009. Unique character instances for crowds. *IEEE Computer Graphics and Applications* 29, 6 (2009), 82–90.

Artur Malinowski and Pawel Czarnul. 2019. Multi-agent large-scale parallel crowd simulation with nvram-based distributed cache. *Journal of Computational Science* 33 (2019), 83–94.

Artur Malinowski, Pawel Czarnul, Krzysztof Czurylo, Maciej Maciejewski, and Pawel Skowron. 2017. Multi-agent large-scale parallel crowd simulation. *Procedia Computer Science* 108 (2017), 917–926.

Koki Nagano, Jaewoo Seo, Jun Xing, Lingyu Wei, Zimo Li, Shunsuke Saito, Aviral Agarwal, Jens Fursund, and Hao Li. 2018. paGAN: real-time avatars using dynamic textures. *ACM Transactions on Graphics (TOG)* 37, 6 (2018), 1–12.

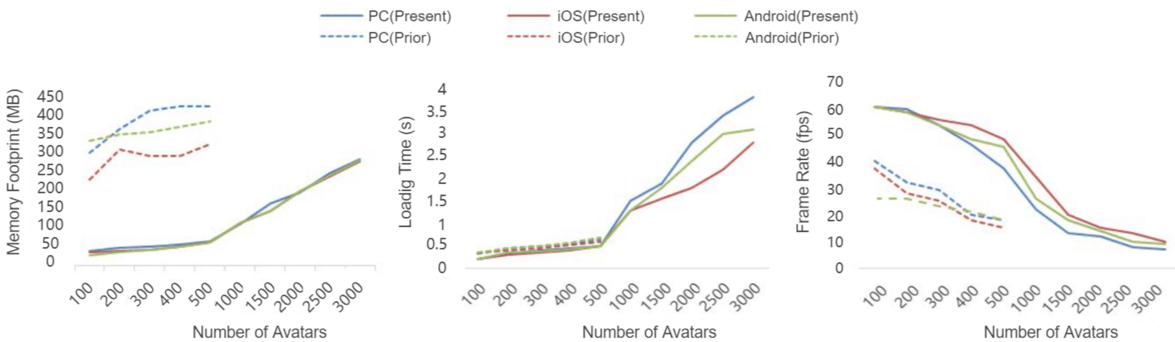

Figure 9: Testing results: from left to right are memory usage, initial loading time and rendering frame rate

Oğuzcan Oğuz, Ateş Akaydın, Türker Yılmaz, and Uğur Güdükbay. 2010. Emergency crowd simulation for outdoor environments. *Computers & Graphics* 34, 2 (2010), 136–144.

Srinivas Peeta, Sushant Sharma, and Yu-Ting Hsu. 2011. Dynamic real-time routing for evacuation response planning and execution. (2011).

Alessandro Pluchino, Cesare Garofalo, Giuseppe Inturri, Andrea Rapisarda, and Matteo Ignaccolo. 2013. Agent-based simulation of pedestrian behaviour in closed spaces: a museum case study. *arXiv preprint arXiv:1302.7153* (2013).

Daniel P Savoy, Marcio C Cabral, and Marcelo K Zuffo. 2015. Crowd simulation rendering for web. In *Proceedings of the 20th International Conference on 3D Web Technology*. 159–160.

Yu Tian, Tian-Shu Zhou, Qin Yao, Mao Zhang, and Jing-Song Li. 2014. Use of an agent-based simulation model to evaluate a mobile-based system for supporting emergency evacuation decision making. *Journal of medical systems* 38, 12 (2014), 149.

Timo von Marcard, Bodo Rosenhahn, Michael J Black, and Gerard Pons-Moll. 2017. Sparse inertial poser: Automatic 3d human pose estimation from sparse imus. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 349–360.

Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. 2011. Realtime performance-based facial animation. *ACM transactions on graphics (TOG)* 30, 4 (2011), 1–10.

Laixiang Wen, Jinyuan Jia, and Shuang Liang. 2014. LPM: lightweight progressive meshes towards smooth transmission of Web3D media over internet. In *Proceedings of the 13th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry*. 95–103.

Fengting Yan, Jinyuan Jia, Yonghao Hu, Qinghua Guo, and Hehua Zhu. 2019. Smart fire evacuation service based on Internet of Things computing for Web3D. *Journal of Internet Technology* 20, 2 (2019), 521–532.

Jinlong Yang, Jean-Sébastien Franco, Franck Hétroy-Wheeler, and Stefanie Wuhrer. 2016. Estimation of human body shape in motion with wide clothing. In *European Conference on Computer Vision*. Springer, 439–454.

Chao Zhang, Sergi Pujades, Michael J Black, and Gerard Pons-Moll. 2017. Detailed, accurate, human shape estimation from clothed 3D scan sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4191–4200.

Hantao Zhao, Tyler Thrash, Mubbasir Kapadia, Katja Wolff, Christoph Hölscher, Dirk Helbing, and Victor R Schinazi. 2020. Assessing crowd management strategies for the 2010 Love Parade disaster using computer simulations and virtual reality. *Journal of the Royal Society Interface* 17, 167 (2020), 20200116.

Xiukai Zhao, Lei Lyu, Chen Lyu, and Cun Ji. 2019. A new parallel simulation method for massive crowd. *Procedia computer science* 147 (2019), 283–287.

Yu-Jun Zheng, Hai-Feng Ling, Jin-Yun Xue, and Sheng-Yong Chen. 2013. Population classification in fire evacuation: A multiobjective particle swarm optimization approach. *IEEE Transactions on Evolutionary Computation* 18, 1 (2013), 70–81.

Kong-jin Zhu and Qin Shi. 2016. Experimental study on choice behavior of pedestrians during building evacuation. *Procedia Engineering* 135 (2016), 207–216.