

Research Article

Cloud Computing Platform for an Online Model Library System

Mingang Chen,^{1,2} Wenjun Cai,³ and Lizhuang Ma¹

¹ Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai 200240, China

² Shanghai Key Laboratory of Computer Software Testing and Evaluating, Shanghai 20112, China

³ Autodesk ACRD, Shanghai 201804, China

Correspondence should be addressed to Mingang Chen; miles_cmg@sjtu.edu.cn

Received 22 January 2013; Accepted 22 March 2013

Academic Editor: Carlo Cattani

Copyright © 2013 Mingang Chen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The rapid developing of digital content industry calls for online model libraries. For the efficiency, user experience, and reliability merits of the model library, this paper designs a Web 3D model library system based on a cloud computing platform. Taking into account complex models, which cause difficulties in real-time 3D interaction, we adopt the model simplification and size adaptive adjustment methods to make the system with more efficient interaction. Meanwhile, a cloud-based architecture is developed to ensure the reliability and scalability of the system. The 3D model library system is intended to be accessible by online users with good interactive experiences. The feasibility of the solution has been tested by experiments.

1. Introduction

It is generally known that the design of 3D virtual scene is the foundation for the production of animations and games, and 3D models are the most important components to construct 3D scenes. Although there have been some quick 3D model construction method [1, 2], the design and modeling of 3D models are extremely time consuming and labor intensive. Therefore, some online 3D model material libraries have been constructed, for example, the Google 3D Warehouse and other commercial 3D libraries like 3D Export, 3D Kingdom, and so forth.

Functions as browsing, storage, sharing, and reusing of models are provided by these web-based online 3D model libraries. Users can reuse models, save their time of modeling, and improve their efficiency of production with the application of 3D model libraries. Therefore, the construction of real-time 3D interactive, full-featured, and reliable system is the trends of the development of 3D model library.

At the same time, image-based proxy is usually used in most of the existing online 3D model libraries to display 3D model, which can only provide limited 3D information and lacking of human-computer interaction. Lately,

the panorama technology [3] provides interactive features for the display of 3D model, but it cannot provide the function of model rotation at any angle and real-time rendering. Google has also launched 3D Warehouse, which provides interactive browsing of 3D models. However, it still has limitations. One is about the rotation of interaction, which tests to be effective only in the horizontal direction. The other is that only Google SketchUp model format is supported.

Web-based 3D model library requires users to contribute new models to the library, while it provides users with model sharing services, which is a typical Web 2.0 pattern. Cloud computing is a fresh paradigm for Web 2.0 that will replace the current dominant web application. So we deploy our Web 3D model library applications onto the cloud, for the advantages of a cloud platform like the scalability of computing and storage, load balancing, failover, and on demand service.

In this paper, a Web 3D model library is designed and implemented based on the cloud computing platform. And a front-end efficient Web 3D and back-end reliable cloud computing solutions are proposed. The paper makes the following contributions. (1) A model format converter is designed to support a wide range of model file formats in the system. (2) The model simplification and optimization

strategies are adopted to improve the efficiency of rich interaction such as scaling and rotation of arbitrary angle. (3) A model size adaptive adjustment module is proposed to fit the model browser quite well. (4) The Web 3D model library is implemented and deployed onto the cloud platform to ensure the stability and scalability of the system.

2. Related Work

Related technologies used in this paper mainly include Web 3D technology and cloud computing technology.

2.1. Web 3D Technology. There have been a large number of Web 3D systems and solutions over the last 20 years. Recently, there have been broadly two categories of Web 3D technologies: one is the browser plug-in-based approach, and the other is a small number of systems, which are trying to integrate the rendering system directly into the browser [4].

In developing content rich Internet applications, Adobe Flash is widely used as a multimedia platform, and it has become a popular solution used to add animation and interactivity to web pages. Flash supports 3D transformations and simulates 3D effects by displaying 2D vector shapes on the screen. According to Adobe, Flash Player has reached over 99% of the Internet users. Therefore, the flash-based technologies are the mainstream Web 3D solution.

Away3D is an open-source 3D graphics engine, written for the Adobe Flash platform in ActionScript 3, and runs in modern web browsers that utilize Adobe Flash Player. It can be used to render 3D models and perform various other 3D computations. It supports hierarchical object transformation with features such as translation, rotation and scaling, rendering of bitmap textures, and real-time lighting effects [5]. In this paper, Away3D technology is used in the Web 3D model library based on the following two reasons. Firstly, 3D effects are highly impressive and the performance is relatively good compared with other Flash 3D engines. The newest version, Away3D 4, has been upgraded to support the latest Flash Player with fully GPU-accelerated graphics. Secondly, Away3D supports multiple external model formats, such as the COLLADA, 3DS, and OBJ.

Other frameworks as Microsoft Silverlight and Oracle Java 3D are also used for developing rich Internet applications in recent years. However, users of Silverlight or Java 3D are much less compared with those using Adobe Flash.

The WebGL solution does not rely on a plug-in and it is a cross-platform web standard for low-level 3D graphics APIs based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces [6]. The development of HTML5 will greatly promote the WebGL solution. Further work has been done with the X3DOM project [7], which overcomes WebGL limitations by providing an adaptation of the rendering technique. However, WebGL is a low-level interface, in which programmers have to deal with every triangle and transformation. Because of the above complex development process, it is not appropriate for the Web 3D model library.

X3D is an XML-based 3D computer graphics file [8]. The X3D specification includes graphics APIs and a full runtime that enables picking, viewing, navigation, scripting, and manipulating the scene graph. X3D supports DOM and Ajax programming and focuses on the visualization of 3D assets. However, so far the mainstream modeling software such as 3ds Max and Maya cannot transform 3D models directly into X3D format.

COLLaborative Design Activity (COLLADA) is an intermediate format with an open standard XML schema whose primary goal is to represent rich data in multiple forms, to exchange digital assets among various graphics software applications [9]. COLLADA defines the format for 3D assets but not their runtime semantics. In this paper, COLLADA is adopted as an interchange model format for the Web 3D model library because of its compatibility with the mainstream 3D modeling software and the well support by Away3D.

2.2. Cloud Computing Technology. In recent years, the cloud computing paradigm is becoming more popular for hosting applications, as it provides businesses with pay-as-you-go computing and storage services [10]. Growing numbers of corporations are involved in this wave to provide public cloud computing services, such as Amazon AWS, Google AppEngine, Microsoft Azure, and Rackspace CloudServer. These cloud providers offer a variety of options in pricing, performance, and feature set. For example, some offer platform as a service (PaaS), where a cloud customer builds applications using the APIs provided by the cloud; others offer infrastructure as a service (IaaS), where a customer runs applications inside virtual machines (VMs), using the APIs provided by their chosen guest operating systems [11].

In this paper, Amazon Web Services (AWSs) [12] are adopted as the cloud computing platform for Web 3D model library mainly due to flexibility of services and its great success in the business compared to Google AppEngine and Microsoft Azure. The Amazon cloud computing platform provides the necessary computing environment, including CPUs, storage, memory, networking, and operating system. The Web 3D model library makes use of three AWSs: Elastic Compute Cloud (EC2), Elastic Block Storage (EBS), and Simple Storage Service (S3). EC2 contains a variety of user selectable virtual machine types, also called instances, that range in computing power and cost. An instance is booted within a few minutes, and the user is given root or administrator access.

3. The Architecture Overview of the Web 3D Model Library System

A classic three-layer architecture is adopted in the Web 3D model library system, that is, data layer, business layer, and presentation layer, as shown in Figure 1.

The data layer deployed in Amazon S3 stands for model files storage. S3 is composed of buckets that are geographically distributed across Amazon's multiple data centers in different locations. So S3 is an extremely reliable persistent storage

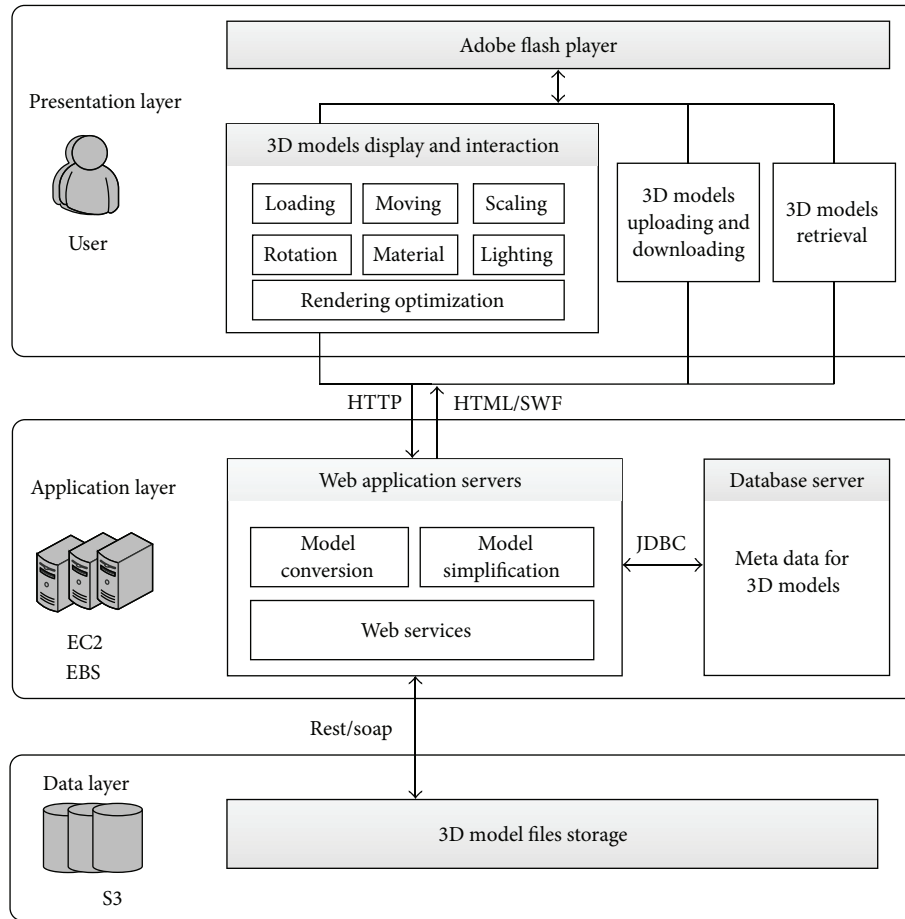


FIGURE 1: The system architecture of Web 3D model library.

system for files storage. When the storage capacity of model files is larger than expectation, storage expansion service of S3 can be easily requested.

Web application servers and models metadata database are deployed on Amazon EC2 that mainly provides business logic of the system, such as user's requests, conversion of model formats, and model's retrieval. Amazon EC2 allows users to deploy scalable resources on demand.

The presentation layer in the user's browser provides 3D model display and interaction, model uploading and downloading functions, and so forth.

This paper will focus on how the system supports 3D model display and interaction, including model format conversion, model mesh simplification, model size adaptive adjustment, and the design of model browser.

4. Model Format Conversion and Model Simplification

The most widespread model files in our model library are 3DS Max and Maya formats. However, these two model formats cannot be rendered in the Away3D engine directly. So in this paper, we design a model format conversion service on the

server side, and this service is for converting various model formats into the COLLADA format.

When models are uploaded to the system, the model format conversion service will first analyze the format of models and then move the COLLADA files into the "COLLADA" folder, while it moves models of other formats into the "NON-COLLADA" folder. The service will get files in the "NON-COLLADA" folder one by one and open it in the relevant modeling software which has installed COLLADA plug-in, and then the format conversion command script is executed. On the server side, the model format conversion service runs periodically. At last, the converted files will be moved to the "COLLADA" folder. Away3D engine will load and render the COLLADA model file for the model's display and interaction.

Furthermore, taking into account complex models may be difficult to real-time 3D interaction, so we also implement a model simplification service on the server side. An iterative decimation algorithm [13] is adopted for the model simplification. This algorithm applies the pair collapse operation iteratively, which replaces two vertices with one, causing neighboring faces to be degenerated. The algorithm defines the quadric error metric for selecting a collapsed pair of vertices from the potential candidates. For a vertex v ,

the quadric error metric is defined as the sum of point-plane distance from v to a set of associated planes, as shown in the following formula:

$$\begin{aligned}\Delta(v) &= \sum_{p \in \text{planes}(v)} (p^T v)^2 \\ &= v^T \left(\sum_{p \in \text{planes}(v)} p^T p \right) v \\ &= v^T Q_v v,\end{aligned}\quad (1)$$

where $\text{planes}(v)$ denotes the triangle faces adjacent to v , and Q_v is the error quadric matrix. In the model simplification process, we assign $\text{planes}(v_k) = \text{planes}(v_i) \cup \text{planes}(v_j)$ by iteratively applying the pair collapse $(v_i, v_j) \rightarrow v_k$.

5. Model Size Adaptive Adjustment

When models are just loaded onto the scene, some are too large to be contained by the screen while others are so small that customers have to pay great efforts to zoom them up. To solve this problem, we have deeply researched into the perspective projection principle of Away3D engine and proposed a method to adjust related parameter automatically and display models of various sizes properly.

The perspective projection matrix P in Away3D can be represented by the following formula:

$$\begin{aligned}P &= \begin{pmatrix} \frac{f}{\text{aspect}} & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & -\frac{N+F}{N-F} & \frac{2NF}{N-F} \\ 0 & 0 & 1 & 0 \end{pmatrix} \\ &= \begin{pmatrix} \frac{N}{W} & 0 & 0 & 0 \\ 0 & \frac{N}{H} & 0 & 0 \\ 0 & 0 & -\frac{N+F}{N-F} & \frac{2NF}{N-F} \\ 0 & 0 & 1 & 0 \end{pmatrix},\end{aligned}\quad (2)$$

where $\text{aspect} = W/H$, $f = 1/\tan(\text{FOV}/2) = N/H$. W and H denote the half of the width and height of the viewport, respectively. N and F denote the distance from the camera to the near plane and the far plane, respectively. FOV is illustrated in Figure 2.

When a 3D point $p(x, y, z)$ is projected onto the projection plane, its 2D coordinate is $p' = (Nx/W, Ny/N)$. So the 3D model bounding box's width and height can be calculated by the following formula:

$$\begin{aligned}\text{model Width} &= \frac{N}{W} (x_{\max} - x_{\min}), \\ \text{model Height} &= \frac{N}{H} (y_{\max} - y_{\min}),\end{aligned}\quad (3)$$

where x_{\max} and x_{\min} denote the maximum and minimum x -coordinate value of all points in a model. Similarly, y_{\max}

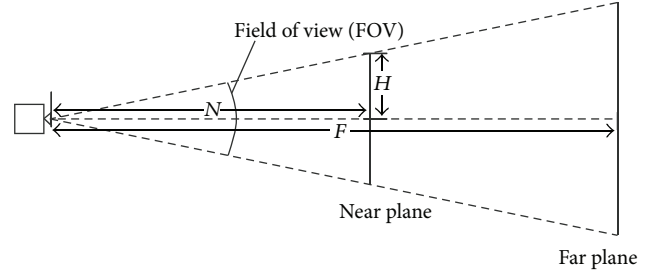


FIGURE 2: An illustration of field of view.

and y_{\min} denote the maximum and minimum y -coordinate value of all points in a model. model Width and model Height denote the model's width and height when it is projected onto the model browser's screen. According to formula (3), we can adjust the distance from the camera to near plane N to a proper value adaptively when models are loaded. Figure 3 compares the display effect of loading models before and after the model size adaptive adjustment.

6. Design of the Model Browser

The model browser based on Away3D is a window for displaying 3D models with multiangle visibility and rich human-computer interaction. The browser supports not only the model rotation, scaling, and translation, but also advanced operations such as material replacement and light source adjustment.

Away3D engine provides rich APIs to process real-time rendering of 3D models. Model's rotation, scaling, and translation operations are implemented by updating the attribute of camera position; model's material replacement and adjustment of light source by updating the attribute of material and light. Away3D engine supports point light, ambient light, or directional light for 3D scene and also specifies ambient shader or diffuse Phong shader, and so forth.

In the following part, we take model's rotation as an example to describe model real-time rendering process in the model browser. Firstly, the model browser program loads 3D model by calling Away3D Collada class and parses model files described with XML. After parsing process, we can get model's information, such as vertex's coordinates and model material. Secondly, the model browser program adds 3D model to the scene and computes the camera's position parameter by formula (3). Thirdly, the event listener functions are registered in the scene to response users real-time interaction. For model's rotation, we just compute the angle of model's rotation. Finally, the model browser program calls Away3D rendering function to realize model's real-time rendering.

7. Application and Performance Analysis

The Web 3D model library system is intended to be accessed by online users and provide users with good interactive experiences, as shown in Figure 4.

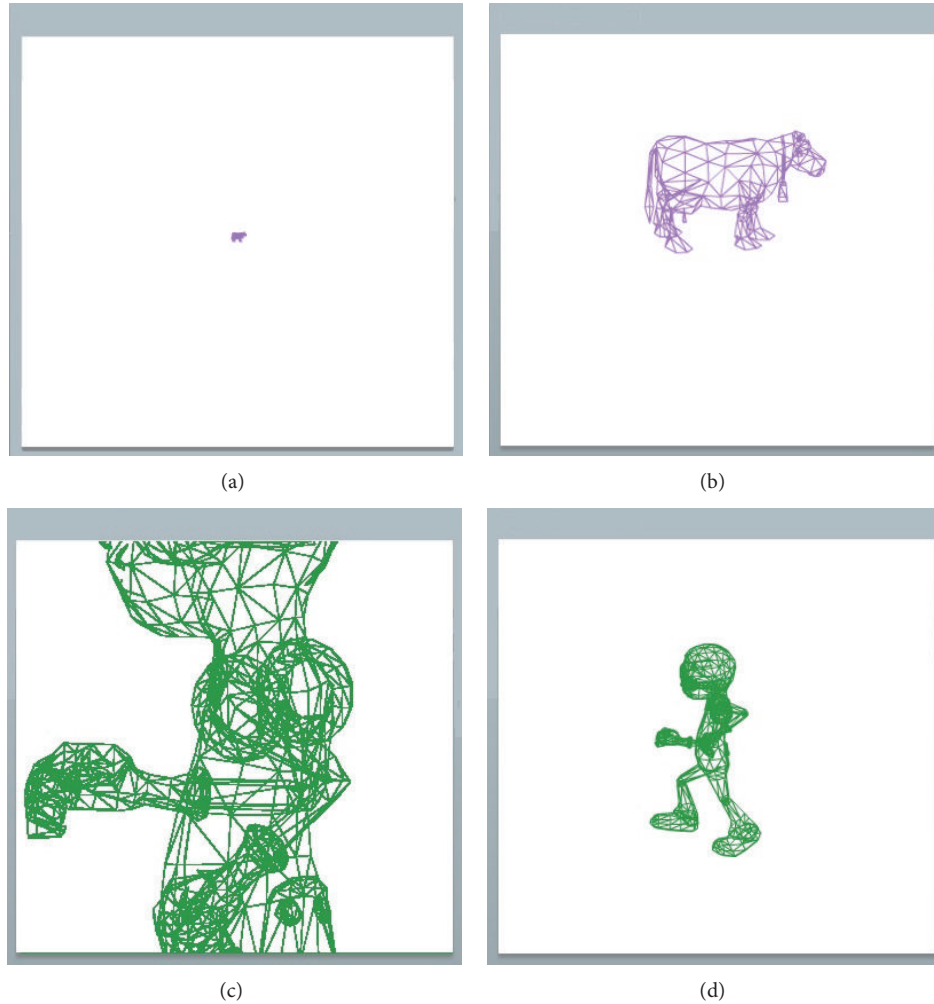


FIGURE 3: (a) and (c) are effects before the model size adaptive adjustment. (b) and (d) are effects after the model size adaptive adjustment.

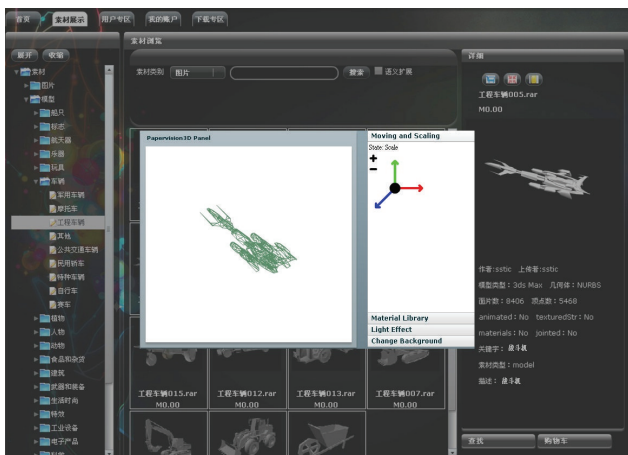


FIGURE 4: Web 3D model library system.

Interactions with 3D models such as rotation, scaling, and translation in the model browser can be realized easily by using a mouse, as shown in Figure 5.

The model browser also supports users to upload images as texture and material and assign them to a model or part of a model, as shown in Figure 6. Figure 6(a) is an original car model, and Figure 6(b) shows that certain parts of car model are selected and are assigned with new materials. Figure 6(c) shows the effect of changing the material of the car model.

Figure 7 shows the light effects of a plane model. The interface is designed for user to change the position of light source, light color, and ambient color. Figure 7(a) shows a plane model without light effects, and Figures 7(b) and 7(c) are the high light and ambient light effects, respectively.

The model rendering time and model browser response time are the important indicators for the model browser's performance. In order to enhance the model browser's performance, in this paper the following optimizing techniques are adopted: (1) choosing suitable Flash stage quality, (2) applying camera's frustum culling process, that is, removing objects that lie outside the viewing frustum from the rendering process [14], and (3) keeping the amount of pixels that need to be rendered as low as possible.

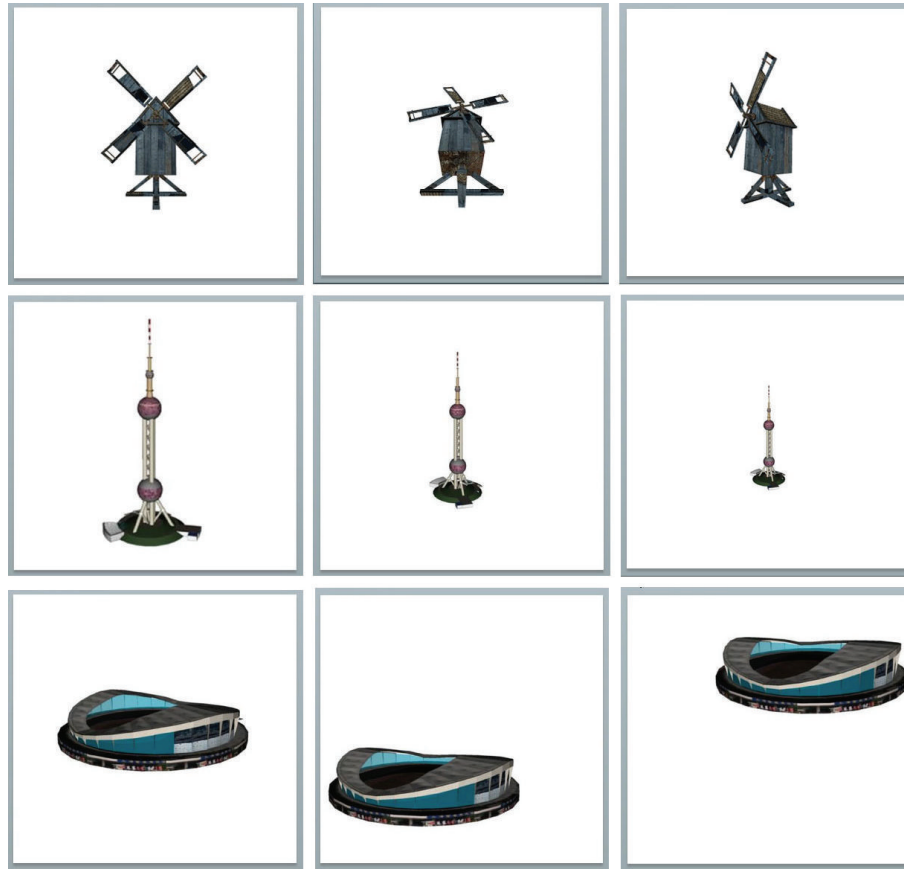


FIGURE 5: 3D model's rotation, scaling, and translation.

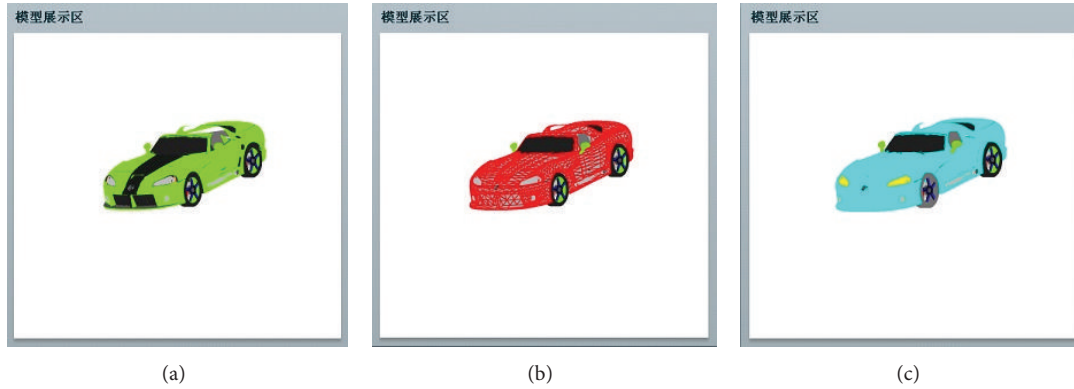


FIGURE 6: The effect of changing the material of a car model.

We choose two models with different complexity to test the model browser's performance, as shown in Table 1. All performance analysis was executed on a laptop with Intel i3 CPU at 2.4 GHz and 4 G memory. From the performance analysis we observe that (1) the more model complex is, the more memory SWF file takes up, and the longer rendering time is, the lower frame rate is and (2) after optimization, memory cost and rendering time are reduced significantly.

8. Conclusion

A web 3D model library system solution and complete design are proposed in this paper. The system is deployed in the Amazon cloud computing platform to ensure its reliability and scalability. In the system, a 3D model conversion service is developed to support a variety of model formats. Furthermore, a model simplification algorithm and a model size

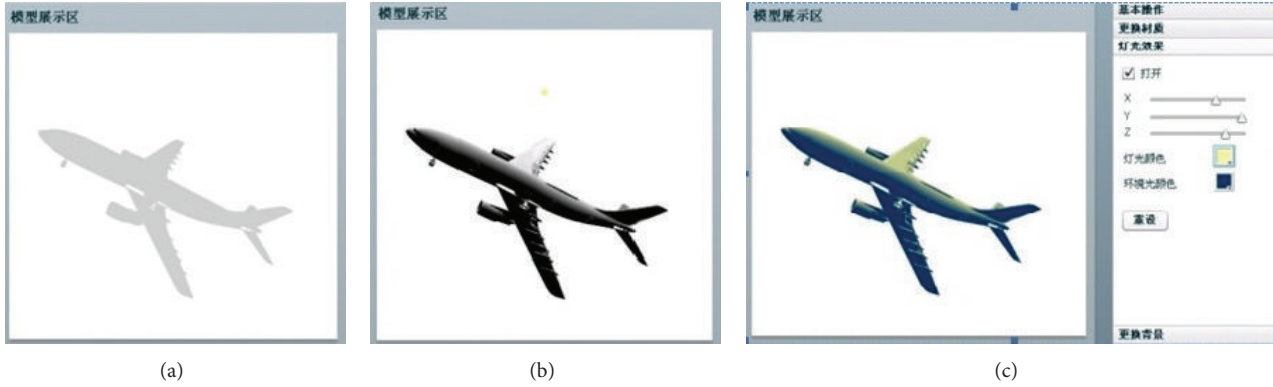


FIGURE 7: The light effects of a plane model.

TABLE 1: Performance analysis of the model browser.

	Windmill model		Plane model	
	Vertex number: 1362		Vertex number: 2724	
	Facet number: 1218		Facet number: 2325	
	Before	After	Before	After
	optimization	optimization	optimization	optimization
Memory (MB)	17.82	12.56	24.38	18.32
Rendering time (ms)	6	4	12	8
Frame rate (fps)	24	32	16	24

adaptive method are designed to improve user's 3D interaction experience. This Web 3D model library solution of design and deployment can be applied to other web applications, such as online virtual tour and design. In further studies, we are planning to design a real-time model simplification method to improve the display accuracy of the complex models.

Acknowledgments

The authors would like to thank professor Jinyuan Jia for his support and valuable suggestions during the project research. This work was funded by the National Natural Science Foundation of China (61133009) and Science and Technology Commission of Shanghai Municipality program (11511500200, 12QB1402300, and 12DZ2290700).

References

- [1] M. Pollefeys, D. Nistér, J. M. Frahm et al., "Detailed real-time urban 3D reconstruction from video," *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 143–167, 2008.
- [2] H. Liu, S. Chen, and N. Kubota, "Special section on intelligent video systems and analytics," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, p. 90, 2012.
- [3] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International Journal of Computer Vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [4] J. Behr, P. Eschler, Y. Jung, and M. Zöllner, "X3DOM—a DOM-based HTML5/ X3D integration model," in *Proceedings of the 14th International Conference on 3D Web Technology (Web3D '09)*, pp. 127–135, June 2009.
- [5] C. Matthew, *Away3D 3.6 Essentials*, Packt Publishing, 2011.
- [6] C. Marrin, *WebGL Specification*, Khronos WebGL Working Group, 2011.
- [7] J. Behr, Y. Jung, J. Keil et al., "A scalable architecture for the HTML5/X3D integration model X3DOM," in *Proceedings of the 15th International Conference on Web 3D Technology (Web3D '10)*, pp. 185–194, July 2010.
- [8] L. Daly and D. Brutzman, "X3D: extensible 3D graphics standard [Standards in a Nutshell]," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 130–135, 2007.
- [9] A. Rémi and P. Tony, "Developing web application with COLLADA and X3D: a whitepaper," 2007.
- [10] M. Cusumano, "Cloud computing and SaaS as new computing platforms," *Communications of the ACM*, vol. 53, no. 4, pp. 27–29, 2010.
- [11] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: comparing public cloud providers," in *Proceedings of the 10th Internet Measurement Conference (IMC '10)*, pp. 1–14, November 2010.
- [12] Amazon Services, <http://aws.amazon.com/>.
- [13] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 209–216, August 1997.
- [14] U. Assarsson and T. Möller, "Optimized view frustum culling algorithms for bounding boxes," *Journal of Graphics Tools*, vol. 5, no. 1, pp. 9–22, 2000.

