International Journal of Advanced Culture Technology Vol.7 No.1 243-247 (2019)
DOI 10.17703/IJACT.2019.7.1.243

IJACT 19-3-33

A Study on 3D File Format for Web-based Scientific Visualization

Geon-hee Lee[1], Jeong-hwan Nam[2], Hwa-seop Han[3], Soon-chul Kwon[1†]

[1]Graduate School of Smart Convergence, Kwangwoon University, Seoul, Korea
[2,3]MANO Landscape Design Group
e-mail : [1]rjsgml5698@gmail.com, [2]manogroup@choi.com, [3]han@vrmano.com, [1†]ksc0226@kw.ac.kr

Abstract

The most commonly used 3D modeling file formats are OBJ (Wavefront file format specification) and STL (STereoLithography). Although they have a common point of view in 3D on the screen, detailed functions are different according to purpose of development. OBJ is the most commonly used 3D file format and STL is mainly used as 3D file format for 3D printing. However, in the field of Scientific Visualization, precise analysis is required. There is a difference in accuracy depending on the type of 3D file format. OBJ and STL are not suitable for delicate surface description because they form meshes in the form of triangular polygons. And if you increase the number of triangle polygons, it will be smoother, but the file size also increases exponentially and causes excessive CPU usage. In contrast, VTK provides a variety of polygon structures, including triangular polygons as well as rectangular polygons and cube polygons. Thus, delicate surface depiction is possible. Delicate surface rendering is possible and file size is not large. This paper describes the concept and structure of VTK. We also compared the load times and file sizes between VTK, STL, and OBJ in the Chrome browser. In addition, the difference in surface rendering ability between VTK, STL, and OBJ is intuitively viewed based on the screen in which each 3D file format is implemented under the same conditions. This study is expected to be helpful for efficient 3D file format for precise implementation of Web - based Scientific Visualization.

Keywords: *WebGL, 3D Modeling, VTK, Scientific Visualization, 3D Viewer*

1. Introduction

The 3D model file format has inherent characteristics depending on the application purpose. The STL format was developed for 3D printing purposes and has no color information. Among the many 3D file formats, the most commonly used OBJ format includes color information, but no animation function. And, Autodesk, a company that provides 3d modeling toolkits, has created a new 3D file format called FBX that adds animation functions and various options.

3D model formats using only triangle polygons have difficulty in describing elaborate surfaces due to structural problems. Therefore, it is not suitable for a 3D file format for Scientific Visualization. The VTX format developed by Kitware solves the structural problem. Figure 1 shows an example of Scientific Visualization configured in VTK format. The detailed structure of the VTK format is divided into two parts: simple legacy and XML. Even in the same VTX format, the supported functions differ according to each structure [1].

Manuscript received: January 30, 2019 / revised: February 20, 2019 / Accepted: March 12, 2019
Corresponding Author: ksc0226@kw.ac.kr
Tel: +82-2-940-8637, Fax: +82-50-4174-3258
Graduate School of Smart Convergence, Kwangwoon University

It is designed to enable more detailed surfaces by refining Polygon objects. This enables precise scientific visualization.

This paper describes the concept and structure of VTK which enables efficient and precise surface description and compares the loading time and file size between 3D file formats in the web environment, Chrome browser. In addition, the difference in surface description between VTK, STL, and OBJ in the same environment was compared [2].

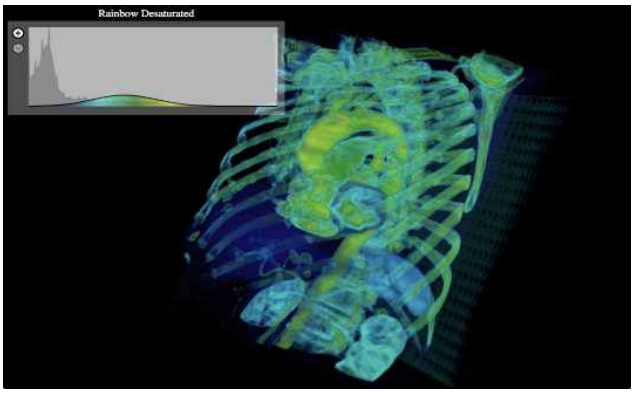


Figure 1. Scientific Visualization using VTK files

2. Background Theory

2.1 Structure of the VTK file format

VTK has two file formats. One is a simple legacy method written in ASCII code. And the other is in the form of an XML file. VTK files in XML format are more flexible than simple legacy methods and have more functions by supporting random access, parallel I / O and portable data compression. In addition, because parallel I / O is supported, efficient scientific visualization is possible when implemented as a multi-threaded computer [3].

The simple legacy method is shown in Fig. 2 (a). (1) is the file version and identifier. (2) is a header and consists of a string ending with the end character \n and a maximum of 256 characters. The Header can be used to describe the data and to include other relevant information. (3) is a file format. The file format is ASCII or Binary. (4) is the data set structure, and the geometry part describes the geometry and topology of the data set. This part begins with a line containing the keyword dataset followed by a keyword that describes the type of dataset. (5) describes the dataset properties and begins with the keyword POINT_DATA or CELL_DATA. Followed by an integer specifying the number of points or cells, respectively.

However, a simple legacy is less flexible than an XML-based file format. As mentioned earlier, the XML format supports arbitrary access, parallel I / O, and portable data compression. And this is preferred over the serial VTK file format. The main goal of configuring the VTK with XML is to facilitate data streaming and parallel I / O. Some features of the format include compression, portable binary encoding, random access, representation of multiple files of fragmented data, and support for new file extensions of various VTK dataset types. XML provides many features, and you can extend the file format with application-specific tags. Figure 2 (b) shows the VTK file structure based on the XML syntax [4].

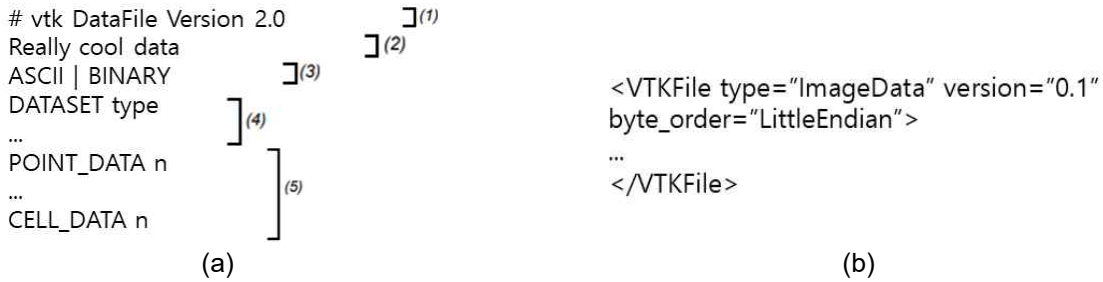


Figure 2. Two Types of VTK files (a) simple legacy Type (b) XML Type

3. Experiment environment

More objective proof of the ease of Scientific Visualization of VTK files is required. The viewer of VTK, STL, OBJ in the same environment was configured in the chrome browser. The important point is that the type and intensity of the light reflected in the three 3D file formats must be the same.
In the experiment, DirectionalLight was applied and the light intensity was set to 1, which is the default value. And the material which shows smoothness and roughness of the surface is made the same kind. The color of the surface is the same when reflecting light. The material type and color are MeshLambertMaterial and white. The 3D JavaScript library applied to this experiment is three.js [5]. And the created JavaScript follows the Vanilla.js syntax. It refers to the original JavaScript syntax, not React.js. For the experiment, the conditions were constructed as shown in Table 1 below.

Table 1. Experiment Environment

Classification	Information		
3D File format	VTK	OBJ	STL
Material(color)	MeshLambertMaterial(0xffffff)		
side	DoubleSide		
3D JavaScript Library	Three.js		
Used JavaScript Library	Vanilla.js(pure JavaScript)		
Light(color)	DirectionalLight(0xffffff)		

The purpose of this paper is to find a suitable format for the implementation of Scientific visualizations among VTK, OBJ, and STL. There are two main types of experiments. Even if the surfaces are fine-grained, there should not be any restrictions on the speed of loading on the Web due to capacity and optimization problems. The web environment in this experiment is a chrome browser and its speed are limited to Fast 3G. Then, the cache is also released so that objective speed verification can be performed. The other one is based on realistic observations of the surface so that the surface description of the 3D file format is visually recognized.

4. Experiments and results

4.1 Performance comparison on the web

Table 2 shows the results and capacity of loading times measured on the same conditions in the Chrome browser (Chrome v.72.0.3626.109) for each 3D file format. The VTK loading time was 17.18 seconds, which was faster than OBJ and STL, and showed a maximum of 17 seconds and a minimum of 10 seconds. Previously, due to the simplicity of the STL file structure, the STL was predicted to be the fastest in terms of the overall loading speed, but VTK showed a great advantage in terms of file size and loading time [6]. The capacity of each 3D file is also 1.4 MB in VTK file format. It is smaller than OBJ and STL and has a maximum file size of 3.5MB and a minimum file size of 0.8MB [7].

Table 2. Performance comparison on the web

Classification	VTK	OBJ	STL
3D File format	Chrome v.72.0.3626.109		
Material(color)	17.18	17.18	17.18
Side	1.4	2.2	4.9
Network Environment	Fast 3G		

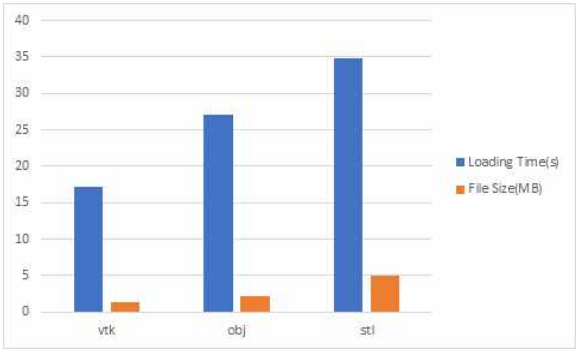
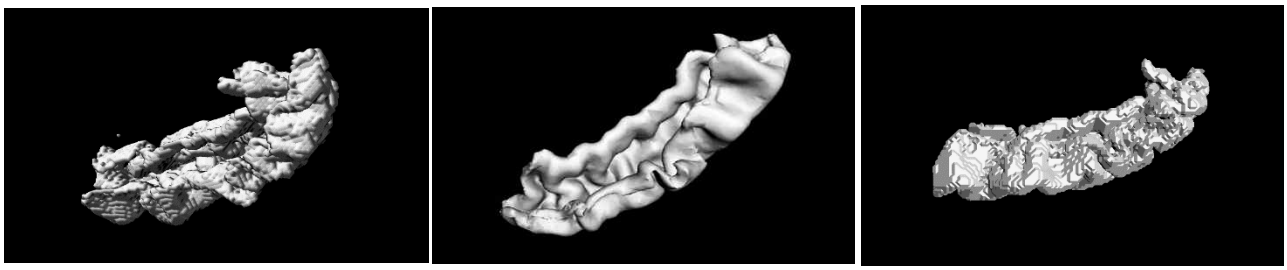


Figure 3. Performance comparison on the web

4.2 Performance comparison on the web

Table 2 shows the benefits of VTK's loading time and file capacity on the Web. In addition, if each 3D file format is implemented in the same environment, the difference can be visually observed more easily [8]. Figure 3 shows the implementation of VTK, OBJ, and STL, which is a fragment of the human brain. Figure 5 (a) shows the implementation with OBJ, and Figure 3 (b) shows the implementation with VTK. Finally, Figure 3 (c) shows the STL implementation. VTK represents the finest detail. As mentioned in the VTK file structure analysis, not only the triangle polygon structure but also the polygon structure is diversified and plays an important role in the detailed description of the surface. The difference between OBJ and STL is not significant, and OBJ is the least recommended 3D file format in terms of loading speed on the web [9]. The VTK file format is best suited for Scientific visualizations due to its elaborate surface representation and small file size [10].



(a) Brain parcel(.OBJ) (b) Brain parcel(.VTK) (c) Brain parcel(.STL)

Figure 4. Brain parcel Implementation (a).OBJ (b).VTK (c).STL

5. Conclusion

In this paper, VTK, OBJ, and STL were compared and analyzed to find the best 3D file format for Scientific Visualization. VTK has been constantly updated and developed by Kitware. Because of the refinement and optimization of the 3D file format, it has a huge effect on Scientific Visualization. Experiments show that VTK is more effective than OBJ and STL for surface rendering and file compression. STL was developed for 3D printers in the 1980s and has not been updated so far, but recently 3D printer market has rapidly expanded and become a widely used 3D file format. However, since it has limited functions to support than VTK, there is not much use except 3D printing.

In the above experiment, the loading speed and the file size on the web are compared and analyzed. As a result, VTK is superior. Also, when implemented in the web environment, the VTK file format is more suitable for Scientific Visualization than the OBJ and STL file formats. Medical science and science are the most important academic disciplines. This paper is expected to be useful for researchers studying efficient 3D file format for Scientific Visualization.

Acknowledgment

This work (Grants No. S2598342) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Ministry of SMEs and Startups in 2018.

References

[1] H. Dong, L. Xia, J. Zhang and A. Cai, "Medical Image Reconstruction Based on ITK and VTK," 2013 International Conference on Computer Sciences and Applications, Wuhan, pp. 642-645, 2013.

[2] S. Jing and S. Jian, "Implementation of Volume Rendering Technique Based on VTK Parallelism in Multi-core Configuration," 2010 Third International Conference on Information and Computing, Wuxi, pp. 318-321, 2010.

[3] J. Rong, C. Zhao, Z. Fan and S. Hu, "Realistic 3D Face Reconstruction Based on VTK and MFC," 2010 International Conference on Optoelectronics and Image Processing, Haikou, pp. 145-149, 2010.

[4] S. Pieper, M. Halle and R. Kikinis, "3D Slicer," 2004 2nd IEEE International Symposium on Biomedical Imaging: Nano to Macro (IEEE Cat No. 04EX821), Arlington, VA, USA, Vol. 1, pp. 632-635, 2004.

[5] Z. Yanyan and R. Xiangjin, "A Study on 3D Visualization System for GoCAD Objects Based on VTK and QT," 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, pp. 47-50, 2016.

[6] P. O'Leary et al., "Enhancements to VTK enabling scientific visualization in immersive environments," 2017 IEEE Virtual Reality (VR), Los Angeles, CA, pp. 186-194, 2017.

[7] Y. Wang, C. Shi and J. Xu, "The research and implementation of a medical image 3-D measurement technique based on VTK," Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology, Harbin, pp. 3174-3177, 2011.

[8] Zhanli Hu and Jin Li, "Study on 3D CT image reconstruction and interactive clipping," 2008 International Conference on Information Technology and Applications in Biomedicine, Shenzhen, 2008, pp. 101-104.

[9] W. J. Schroeder, L. S. Avila and W. Hoffman, "Visualizing with VTK: a tutorial," in IEEE Computer Graphics and Applications, vol. 20, no. 5, pp. 20-27, 2000.

[10] Yuxuan Huang, Zhongpan Qiu and Zhijun Song, "3D reconstruction and visualization from 2D CT images," 2011 IEEE International Symposium on IT in Medicine and Education, Guangzhou, pp. 153-157, 2011.