

# Plateforme Web pour le streaming de données 3D compressées

English title

Guillaume Lavoué et Florent Dupont

**English Abstract**—This paper presents a technical solution for remote streaming and visualization of compressed 3D data, which runs natively in a Web browser without any plugin installation. Our approach leans upon three strong features: (1) a dedicated progressive compression algorithm for 3D graphics data with colors producing a binary compressed format which allows a progressive decompression with several levels of details; (2) the introduction of a JavaScript halfedge data structure allowing complex geometrical and topological operations on a 3D mesh; (3) the JavaScript implementation of the decompression scheme allowing 3D data streaming in a Web browser. Experiments illustrate the significant gain of our approach regarding the classical uncompressed 3D data transmission, in terms of latency and quality of user experience.

## 1 INTRODUCTION

Technological advances in the fields of telecommunication, computer graphics, and hardware design during the two last decades have contributed to the development of a new type of multimedia: three-dimensional (3D) graphics data. The visualization of 3D content through the Web is now possible thanks to specific formats like the Virtual Reality Modeling Language (VRML) (now replaced by X3D), technologies like the very recent WebGL, which makes the GPU controllable by JavaScript, and norms like HTML 5. The Web3D concept (i.e. communicating 3D content on the Web) is seen as the future of the actual Web 2.0, and is supported by many organizations like the W3C and the Web3D consortium. However, as observed by Di Benedetto et al. [1], the delivery and visualization of 3D content through the Web has come with a considerable delay with respect to other digital media such as images, videos and sound, mainly because of the higher requirements of 3D graphics in terms of computational power. First systems used Java Applets or ActiveX controls to expose 3D data on a web browser, however very recently more powerful tools have been introduced like WebGL [2] that will probably boost the use of 3D data on the Web. A lot of industries have interest in providing 3D content through the Web, including online video games (to represent virtual worlds), 3D design or e-business companies; for instance 3D model designers (both for mechanical engineering or artistic purposes) have interest to expose the full 3D models of their

products instead of 2D screenshots. Moreover like existing huge repositories of pictures (e.g. Flickr) or videos (e.g. YouTube), community Web 3D model repositories are now appearing, such as Google 3D warehouse.

An efficient system for interactive remote visualization of large 3D datasets needs to tackle the following technical issues:

- 1) Removing the latency; in most of existing systems, 3D data are fully loaded in a uncompressed form, hence there is latency before visualization.
- 2) Allowing the adaptation of the level of details to different transmission networks and client hardwares, in order to allow a good frame-rate even in case of low-power devices like smartphones.
- 3) Providing an easy access and portability among all possible visualization clients.

Issues (1) and (2) can be resolved by the use of progressive compression techniques [5]; indeed, progressive compression allows to achieve high compression ratio (and thus fast transmission) and also to produce different levels of details (LoD), allowing to adapt the complexity of the data to the remote device by stopping the transmission when a sufficient LoD is reached. Moreover users are able to quickly visualize a coarse version of the 3D data first, instead of waiting for full objects to be downloaded before they can be displayed. Figure 1 illustrates different levels of details for the power plant tank 3D model. These functionalities are able to neutralize the time latency even for huge data and make possible real-time interactions (i.e. high frame rate) even for mobile devices.

Concerning issue (3), the most accessible system is a Web platform; indeed like stated in the recent study

---

• Guillaume Lavoué: Université de Lyon, CNRS, LIRIS, INSA-Lyon  
E-mail: glavoue@liris.cnrs.fr.  
• Florent Dupont: Université de Lyon, CNRS, LIRIS, Université Lyon 1  
E-mail: fdupont@liris.cnrs.fr.

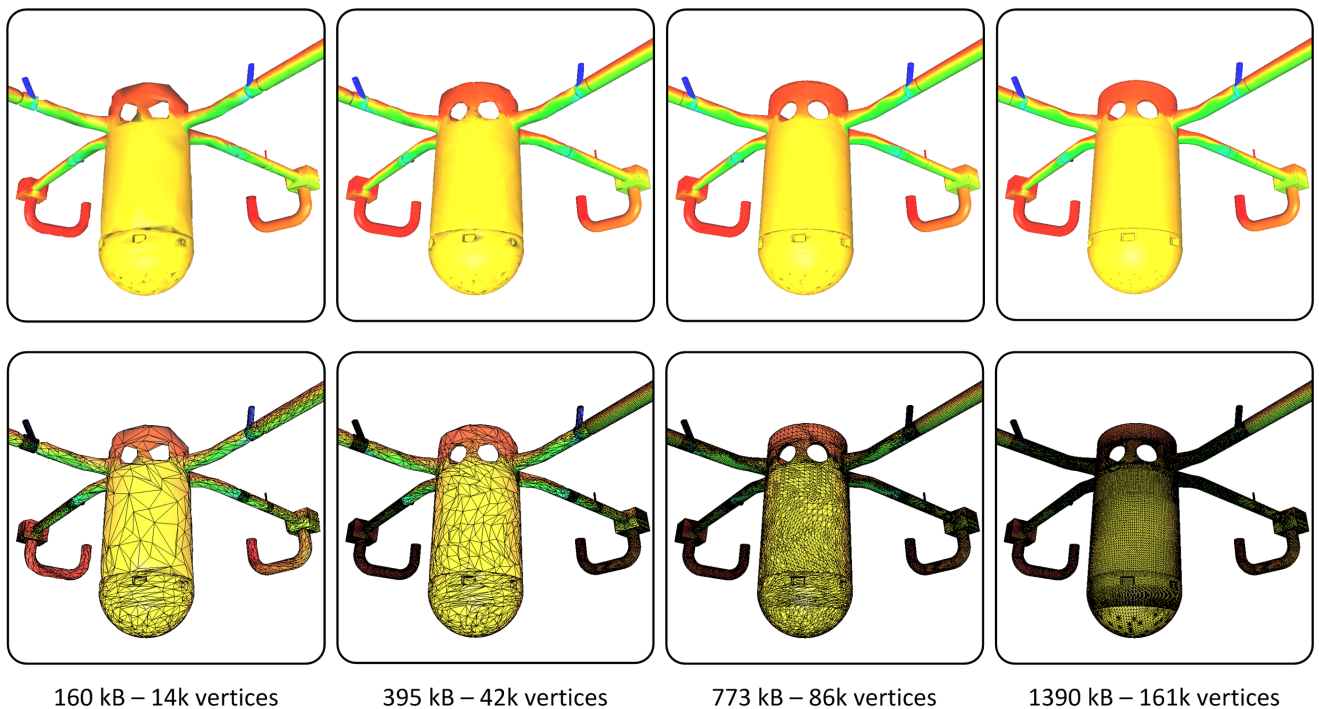


Fig. 1. Progressive decoding of the compressed P3DW file corresponding to the *Tank* model (with and without wireframe). From left to right: 12%, 28%, 56% and 100% of the stream are respectively decoded. Such progressive decoding allows to stream the data so as to obtain very fast a good approximation of the model, moreover it allows an adaptation to the client device hardware (for a high performance workstation the full resolution model can be loaded and visualized interactively but in case of a smartphone, a low resolution version has to be preferred). The original ASCII OFF file size is 12762 kB.

from Mouton et al. [4], Web Applications have major benefits compared to desktop applications. Firstly, Web browsers are available for all mainstream platforms including mobile devices, and secondly the deployment of Web Applications is straightforward and does not require the user to install or update softwares or libraries other than the browser.

## 2 OUR CONTRIBUTIONS

We introduce a technical solution for Web-based remote 3D streaming and visualization, which tackles the three issues mentioned above. Our system runs natively in a Web browser without any plugin installation and leans upon three strong features: (1) a dedicated progressive compression algorithm for 3D graphics data with colors, producing a binary compressed .P3DW file which allows a progressive decompression with several levels of details; this algorithm is Web-based adaptation of our progressive algorithm described in [3]; (2) the introduction of a Javascript halfedge data structure allowing geometrical and topological operations on a 3D mesh; (3) the multi-thread JavaScript implementation of the associated decompression scheme, using WebGL, allowing 3D data streaming in a Web browser.

## 3 CONCLUSION

Our approach brings a clear gain in term of quality of user experience by removing the latency and providing very fast a good approximation of the 3D model even for huge data.

One remaining critical issue for the practical industrial use of Web 3D data streaming is the question of the intellectual property protection; indeed during its transmission or visualization the 3D content can be duplicated and redistributed by a pirate. This issue can be resolved with the use of watermarking techniques; such technique hides a secret information in the functional part of the cover content (usually the geometry in case of 3D data). We plan to integrate such watermarking algorithm in the next version of our Web platform.

## REFERENCES

- [1] M. Di Benedetto, F. Ponchio, F. Ganovelli, and R. Scopigno. SpiderGL: a JavaScript 3D graphics library for next-generation WWW. In *Web3D*, pp. 165–174, 2010.
- [2] Khronos. WebGL - OpenGL ES 2.0 for the Web, 2009.
- [3] H. Lee, G. Lavoué, and F. Dupont. Rate-distortion optimization for progressive compression of 3D mesh with color attributes. *The Visual Computer*, 28(2):137–153, May 2012.
- [4] C. Mouton, K. Sons, and Ian Grimstead. Collaborative visualization: current systems and future trends. In *Web3D*, 2011.
- [5] J. Peng, C.-S. Kim, and C.-C. J. Kuo. Technologies for 3D mesh compression: A survey. *Journal of Visual Communication and Image Representation*, 16(6):688–733, 2005.