Bartosz SAWICKI, Bartosz CHABER

Warsaw University of Technology

3D Mesh Viewer Using HTML5 Technology

Abstract. *The effective visualization of 3D meshes based on the build-in features of the web browser supporting HTML5 (Canvas, WebGL) standards are presented. The algorithms for progressive streaming mesh over Internet network are presented. It has been demonstrated that using described software, 3D models could be easily inspected on any modern, mobile device (Internet tablets, smart-phones or netbooks).*

Streszczenie. *Artykuł prezentuje wydajną metodę wizualizacji siatek trójwymiarowych przy pomocy mechanizmów wbudowanych w przeglądarki internetowe obsługujące standard HTML5 (Canvas, WebGL). Przedstawiono algorytmy progresywnego przesyłania siatki poprzez sieć internet. Zademonstrowano sposób, który pozwala na łatwą, interaktywną pracę z obiektami 3D na współczesnych, przenośnych urządzeniach takich jak tablety, smartfony, czy netbooki.* ***Wykorzystanie standardu HTML5 do wizualizacji siatek trójwymiarowych***

Keywords: wizualizacja trójwymiarowa, Canvas, WebGL
Słowa kluczowe: 3D visualization, Canvas, WebGL

Introduction

For several years a three dimensional reality is widely present in the computer science industry. The most popular applications are related with games placed in the virtual, realistic worlds. But this paper is not about 3D games. It is addressed to the applications for visualization of scientific simulations results. With instantly growing computational power, numerical simulations is becoming the most popular scientific tool. For the same reason visual inspection of 3D models is every day task for many researchers.

The OpenGL is well established programming standard how to efficiently describe 3D scene, and render it on computer screen. The OpenGL works as a part of classic operating system, where standalone application has direct access into the hardware. Nowadays software is more often served as a service (SaaS) using Internet network. In this paper we follow this modern trends. We will discuss visualization of 3D objects inside the Internet browser window [1].

One of the first technologies which were designed for 3D visualization inside the browsers was VRML (year 1995). This text based meta-language inspired by HTML was proposed for specifying 3D scenes in terms of geometry and material properties. For the rendering in the web browser it was required to install a platform specific plug-in. Currently X3D is the successor to the VRML.

Today, the Java Applets are probably the most popular solution for 3D in a web browser. An applet is a small application written in Java which is embeded inside the HTML code of a webpage. To run it special plugin is required as well as Java Virtual Machine. The implementation of the JVM on all the operating systems made Java applets ubiquitous. Java has build-in features for binding to OpenGL called JOGL which gives control on the 3D graphics hardware.

HTML5 3D viewer

The HTML5 is new version of HTML which is core language of WWW network. Even though standard is still under development (on June 2011), it is supported by majority of modern browsers. HTML5 introduced several new technologies, which are designed to eliminate last differences between desktop and web applications. The graphic interface has been benefited by Canvas and WebGL. These new components are closely related. Canvas is general element for 2D drawings, which is extended by 3D context in WebGL. All of features are programmable from Javascript language.

The WebGL standard is managed by Kronos Gruop [2]. The library enables direct acceleration of 3D objects using OpenGL ES. Programming OpenGL transformations could be time consuming, so plenty of middle-ware project in Javascript were established to facilitate 3D operations. Such as C3DL [4], Processingjs [5] or WebGLU.

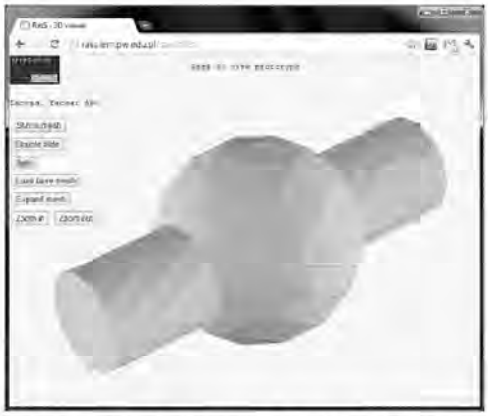

Fig. 1. Screenshot of 3D viewer prototype inside 3D Chrome browser window.

In presented project (see Fig. 1) we decided to use three.js library [3] which supports rendering for GPU accelerated hardware, but if such device is absent rendering will be performed as a pure Canvas element. That way we could get more flexible software. Another important advantage of our project is utilization of progressive mesh loading. As shown in Fig. 2 initial mesh is very coarse, then simultaneously with the user interaction, mesh refinement is loading.

Fig. 2. Loading of progressive mesh. On the left initial tetrahedron, which is displayed and simultaneously refined to the original mesh (on the right).

BIBLIOGRAPHY

[1] E. Pinto, G. Amador, A. Gomes: A graphics library for delivering 3D contents on Web browsers Digital Content, Multimedia Technology and its Applications (IDC), 2010 6th International Conference on Seoul, 16-18 Aug. 2010
[2] Khronos Group, Webgl - opengl es 2.0 for the web, 2009, http://www.khronos.org/webgl/
[3] Mr. Doob, three.js, Javascript 3D Engine, https://github.com/mrdoob/three.js, 2010-2011
[4] C3DL - The Canvas 3D JS Library, 2011, http://www.c3dl.org
[5] John Resig, processing.js, http://processingjs.org

Authors: Bartosz Sawicki, Bartosz Chaber, Institute of Theory of Electrical Engineering, Measurement and Information Systems, Faculty of Electrical Engineering, Warsaw University of Technology, ul. Koszykowa 75, 00-662 Warszawa, Poland, email: sawickib@iem.pw.edu.pl