Real-Time Web GIS Analysis Using WebGL

Michael Auer, University of Heidelberg, Germany

ABSTRACT

Parallel processing methods in Geographic Information Systems (GIS) are traditionally used to accelerate the calculation of large data volumes with sophisticated spatial algorithms. Such kinds of acceleration can also be applied to provide real-time GIS applications to improve the responsiveness of user interactions with the data. This paper presents a method to enable this approach for Web GIS applications. It uses the JavaScript 3D graphics API (WebGL) to perform client-side parallel real-time computations of 2D or 2.5D spatial raster algorithms on the graphics card. The potential of this approach is evaluated using an example implementation of a hillshade algorithm. Performance comparisons of parallel and sequential computations reveal acceleration factors between 25 and 100, mainly depending on mobile or desktop environments.

Keywords: Canvas, Graphic Processing Unit (GPU) Computation, Hillshade, Hypertext Markup Language 5 (HTML5), Spatial Analysis, Web Geographic Information Systems (GIS), Web Graphic Library (WebGL)

INTRODUCTION

The term real-time Web GIS can refer to different kinds of applications. It can be understood as a system visualizing in real-time measured sensor data, like GPS positions or in real-time generated data by human editors, e.g., in collaborative disaster management systems. In the scope of this paper it refers to an almost immediate visualization of GIS analysis results in order to provide a more interactive and responsive user experience, irrespective of the creation time of the input data. Reducing waiting times in web applications is a general aim. Besides that, a real-time linkage between user behavior and analysis results can also be used to enable visual data exploration in data mining processes. Two aspects of the presented approach contribute to the reduction of time between user request and presentation of the results. Spatial analysis execution in the web browser and doing this in an accelerated way using parallel computation on the graphics processing unit (GPU) instead of classic serial computation on the central processing unit (CPU).

In traditional web based GIS applications it is common to use a client-server architecture where the computational analysis part is done on the server side while the web client is just used as a user interface for requesting those computations and to display the processed results. The reason for this is the lack of graphic manipulation capabilities of current web standards. This restriction will change with future web standards and browser releases.

DOI: 10.4018/ij3dim.2012070105
Currently ongoing developments in web technology, like HTML5 (Hickson, 2012; WHATWG, 2012) and WebGL (Khronos Group, 2011a), offer great potential for the future development of more interactive, responsive, efficient and mobile Web GIS applications. This includes the usage of 2D, 3D and even temporal and animated content without the need of any third party plugins.

The HTML5 specification introduces a new element type called canvas. For the first time, since the invention of the first widely used graphical web browser “Mosaic” in 1993 (Peterson, 2003), the canvas element enables web developers to not only present images, but manipulate images and also 3D scenes within the website using JavaScript. There are two different possibilities to manipulate graphics within a canvas element. One is the built in 2D drawing API for raster images and the other one is the WebGL API which can be used to create interactive 3D scenes with textured objects consisting of points, lines or triangles. Those two APIs are also referred to as “2d”- or “webgl”-contexts of the canvas element. The WebGL API has a close connection to the graphics card. Its hardware is designed to rapidly project and rasterize 3D vector objects to the canvas plane on the screen. It can do such transformations very fast because it uses hundreds of specialized microprocessors in parallel. The ability to compute self-defined functions in parallel and output a raster graphic on a webpage as a result is the key property of the WebGL 3D-Technology, which in this paper will be exploited to outline the possibilities for extremely performing client side 2.5D Web GIS analysis using the example of a hillshade computation. The application of this approach opens up new perspectives on user interaction and interface design for the experience of in real-time responsive analysis results. A demonstration of the method can be evaluated on http://webgl.uni-hd.de/realtime-WebGIS.

The following sections provide an overview on related work about parallel computation approaches of GIS analysis. A short introduction on the basics of 3D rendering and a detailed description how to use WebGL for the proposed method follows. Issues concerning the conversion of geospatial raster data to web compatible formats are addressed. An example hillshade algorithm is described and used to demonstrate and measure the performance of the approach. Furthermore some limitations of WebGL based spatial raster processing are discussed.

RELATED WORK

Performing analysis as fast as possible is an intrinsic aim of any GIS application as the amount of available spatial data is growing fast and depending on the application domain sophisticated time-consuming algorithms are used. Therefore the approach to parallelize GIS processing has been applied in many different contexts (Healey, 1996). Despite the potential of the approach, parallel GIS software has not yet become an everyday tool to GIS analysts and most commercial software doesn’t support it as common parallel architectures depend on specific infrastructure like high performance computing clusters or Grid computing infrastructure (Lanig & Zipf, 2009). A recent trend to speed up desktop GIS analysis has been enabled due to the development of frameworks for general-purpose computing on graphics processing units (GPGPU). A major drawback of these frameworks is that they are platform dependent. CUDA is the framework for Nvidia products (NVIDIA, n.d.), whereas APP SDK is the one for AMD (n.d.). The non-profit industry consortium Khronos Group, which also released the specification for WebGL (Khronos Group, 2011a), has developed a platform independent standard called OpenCL (Khronos Group, 2011b) to overcome these technology barriers. Several research projects have already illustrated the applicability to GIS with particular focus on raster operations. Huraj et al. (2010) are using GPGPU computation for an inverse-distance-weighting (IDW) interpolation to model snow coverage from meteorological data of 20 years. They could gain a speedup of 6.7 times compared to an implementation written in the C programming language. Xia
Innovative ICT Applications in Transport and Logistics: Some Evidence from Asia
www.igi-global.com/chapter/innovative-ict-applications-transport-logistics/70555?camid=4v1a