Computers & Geosciences 75 (2015) 57–65

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

A parallel algorithm for viewshed analysis in three-dimensional Digital Earth

Wang Feng [a], Wang Gang [a,*], Pan Deji [a], Liu Yuan [a], Yang Liuzhong [b], Wang Hongbo [c]

[a] State Key Laboratory of Remote Sensing Science, The Institute of Remote Sensing and Digital Earth, Chinese Academy of Science, Beijing 100101, China
[b] Urban-Rural Planning Management Center, Ministry of Housing and Urban-Rural Development of the People's Republic of China, Beijing 100835, China
[c] Managers Training Institute, China National Petroleum Group, Beijing 100096, China

ARTICLE INFO

Article history:
Received 17 April 2014
Received in revised form
16 October 2014
Accepted 29 October 2014
Available online 6 November 2014

Keywords:
Viewshed analysis
Digital Earth
GPU
Visualization
Vertex shader
Pixel shader

ABSTRACT

Viewshed analysis, often supported by geographic information systems, is widely used in the three-dimensional (3D) Digital Earth system. Many of the analyzes involve the siting of features and real-timedecision-making. Viewshed analysis is usually performed at a large scale, which poses substantial computational challenges, as geographic datasets continue to become increasingly large. Previous research on viewshed analysis has been generally limited to a single data structure (i.e., DEM), which cannot be used to analyze viewsheds in complicated scenes. In this paper, a real-time algorithm for viewshed analysis in Digital Earth is presented using the parallel computing of graphics processing units (GPUs). An occlusion for each geometric entity in the neighbor space of the viewshed point is generated according to line-of-sight. The region within the occlusion is marked by a stencil buffer within the programmable 3D visualization pipeline. The marked region is drawn with red color concurrently. In contrast to traditional algorithms based on line-of-sight, the new algorithm, in which the viewshed calculation is integrated with the rendering module, is more efficient and stable. This proposed method of viewshed generation is closer to the reality of the virtual geographic environment. No DEM interpolation, which is seen as a computational burden, is needed. The algorithm was implemented in a 3D Digital Earth system (GeoBeans3D) with the DirectX application programming interface (API) and has been widely used in a range of applications.

1. Introduction

Digital Earth, as a multi-resolution, three-dimensional (3D) representation of the planet, allows users to locate, visualize, and interpret vast amounts of geo-referenced information. In addition to its primary functions of displaying the world's terrain and remote sensing imagery, one of the most valuable functions of Digital Earth is spatial analysis, which is the primary function of 3D Geographic Information Systems (GIS) (Shi and Liu, 2005). Viewshed analysis is one type of these spatial analyzes. This process involves predicting the total area that is visible from a single point or multiple points (Zhou and Liu, 2006). Viewshed analysis has been used in a wide range of applications, including locating telecommunication relay towers (De Floriani et al., 1994), locating wind turbines (Kidner et al., 1999), protecting endangered species (Camp et al., 1997), and searching for archeological locations (Lake et al., 1998).

A viewshed, in the virtual terrain environment, is a collection of points that are visible from a specific point. Viewshed calculations are potentially time consuming, mainly because extensive interpolation is necessary when using a gridded digital elevation model (DEM) due to the complex terrain model and the complicated geometric features. In addition to the time complexity, multi-point viewshed analyzes and the integration of viewshed calculations with Digital Earth to speed up the calculation of dynamic viewshed analyses are also major challenges to researchers. Therefore, much work has been conducted to develop an efficient viewshed algorithm. Section 2 provides a brief overview and discussion of previous work on viewshed analysis. These previous studies are valuable and can help improve the efficiency of viewshed analysis.

Graphics processing units (GPUs) have been recently used in a large number of applications because they can provide substantial computational power at an affordable cost, and their programmability has also improved (Owens et al., 2007). The parallel property of GPUs has been increasingly utilized to improve computational performance. However, traditional viewshed analysis algorithms, which interpret the viewshed as a series of

* Corresponding author.
E-mail addresses: luoying_gis@126.com (W. Feng), wg638@126.com (W. Gang).

http://dx.doi.org/10.1016/j.cageo.2014.10.012

intervisibility calculations to all vertices of the DEM based on line-of-sight (LOS), are mainly executed on a computer's central processing unit (CPU) without making full use of high-end GPUs. With the development of GPUs, some GPU-based viewshed algorithms have been proposed in recent three years.

The present research focuses on solving the performance issues described above and providing users with an effective, real-time, parallel, and GPU-based viewshed analysis algorithm. This algorithm has been applied in our 3D Digital Earth application (GeoBeans) with good results.

2. An overview of previous work

Viewshed analysis, as an important branch of 3D spatial analysis, has received increasing attention among researchers. Related academic, conference, and research monographs regarding this approach are gradually increasing (Han, 2011). Current research mainly focuses on viewshed analyzes in terrain models whose data structure is a DEM or triangulated irregular network (TIN). Other previous studies have investigated building-blocks analysis and sunlight analysis (Ying, 2005).

The basic algorithm for generating a viewshed from raster elevation data, known as intervisibility, is based on the estimation of the elevation difference of intermediate pixels between the viewpoint and target pixels. A line segment between a viewpoint O and a target point A, which makes up the LOS, is created to determine the visibility of target point A. Moving along the line segment OA and testing all of the points along this line, target point A is visible only if all of the points on OA have an elevation higher than the elevation of the corresponding point on the terrain. Otherwise, target point A is invisible from the viewpoint (see Fig. 1). The LOS computation is repeated for all target points within the viewshed range of the viewpoint during viewshed analysis. This process is extremely time consuming, and its time complexity is expressed as $O(n^2)$.

The brute-force algorithm described above is simple but computationally intense. A variety of algorithms have been developed to speed up these calculations. De Floriani et al. (1994) proposed an algorithm, named the key slope method that is a huge improvement over LOS. This method continually computes the slope along the sightline and updates the maximum slope. The slope of the current point is compared with the max slope to determine the visibility. Unnecessary computations are greatly reduced, resulting in a time complexity of $O(n)$ using this method. A new, double increment method is presented by Yin shen to speed up the calculation. The accuracy, indeterminacy, and invariance of viewsheds are also discussed (Ying, 2005). Liu et al. (2010) proposed an improved algorithm by using the slope and elevation between the target pixel and viewshed point to reduce the required computation.

Yanlan (2001) introduced a new algorithm to determine viewshed without using sightline, named the reference plane method, and this algorithm is considered more effective than the LOS algorithm because no DEM interpolation is needed. A reference plane, based on the spatial topological relation of the viewpoint and the target point, is generated to calculate the visibility. Unlike the LOS algorithm, this method generates viewshed without redundant computation. However, this technique is limited to DEMs and not suited to calculating a viewshed from varying resolutions.

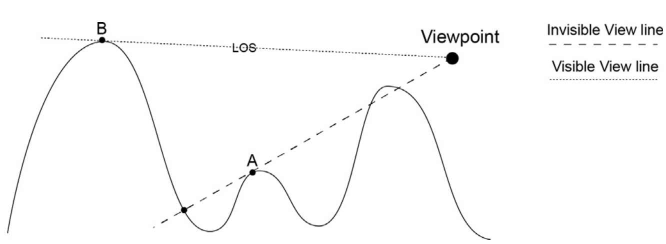

Fig. 1. Overview of the traditional LOS algorithm showing a comparison of the height of the target point with other points along the sight line to determine the visibility. This process is computationally intense. Point B is visible from the viewpoint while point A is invisible.

Generally, GPUs are designed to exploit data parallelism. It has been reported that GPUs can achieve 10 times more floating-point operations per second (FLOPS) than CPUs (Govindaraju et al., 2006). With the rapid development of modern GPUs, transferring traditional algorithms that were previously executed on CPUs to GPUs is becoming increasingly popular. This technique enables GPUs to process repeated computing tasks to speed up the computation. Chen et al. (2010) implemented a rapid contour-line-extraction algorithm by using traditional methods on GPUs. Parallel processing techniques have also been applied to improve computational performance in viewshed analyzes (Mills et al., 1992, Gao et al., 2011, Zhao et al., 2013). All these methods proposed strategies for implementing traditional LOS-based interpolation viewshed algorithm with NVIDIA CUDA. Although made a progress in efficiency, those methods are limited to regular square grids (RSGs) and not suitable for triangulated irregular networks (TINs), neither for complicated scene with geometry features. Fang et al. (2011) introd Fang uced a real-time parallel algorithm for viewshed analysis known as shadow map-based algorithm. This algorithm is executed on GPUs and uses a depth buffer to store the pixel's minimum depth. Comparing the depth of the current pixel with the depth of a corresponding pixel recorded on the depth buffer, one pixel is visible only if its depth is lower than the minimum depth. This method has an advantage of avoiding most of the computation on a CPU and without consideration of the data-structure and DEM resolution. Nevertheless, one of the disadvantages of this method is its low accuracy. In Fang's method, the size and depth of the shadow map determine the quality of the final results, and low-accuracy areas are usually visible as aliasing or shadow continuity glitches.

3. GPU-based parallel algorithm for viewshed analysis

3.1. Principle

Our parallel algorithm takes a new approach to simulating viewshed analysis by creating occlusive volumes to shield the geometric features in the neighborhood of the viewpoint. In contrast to the proposed GPU-based algorithm (Yanli Zhao et al., 2013), this method avoid the interpolation operation which is time consuming. The surface of the geometry features is used to display the analysis result, called the receiver. Occlusive volumes, known as caster, are generated according to the position of the viewpoint and geometric feature outlines by casting the feature's outline along the sightline to infinity. Although this process still utilizes a sightline to generate occlusive volumes, it differs from the LOS method because DEM interpolation is not required and calculation redundancy is extremely reduced. All of the geometric features, including terrain, models, and trees, in the specific space of the required viewpoint can be used for both the caster and receiver. Therefore, our proposed algorithm performs well in complicated three-dimensional scenes, whereas traditional methods do not. Users only have to add the updated feature as a new caster when the scene is updated, and no changes to the code logic are required. With this prerequisite, our algorithm can conveniently and efficiently simulate viewshed calculations by transforming this process to identify and label the pixels that are within the