Earth Science Informatics (2021) 14:619–632
https://doi.org/10.1007/s12145-020-00545-7

RESEARCH ARTICLE

PDERL: an accurate and fast algorithm with a novel perspective on solving the old viewshed analysis problem

Chuanjun Wu[1] · Lingxiao Guan[1] · Qing Xia[1] · Gang Chen[1] · Baohong Shen[1]

Received: 26 April 2020 / Accepted: 25 October 2020 / Published online: 29 October 2020

Abstract

Viewshed analysis based on the regular grid digital elevation model (DEM) is one of the basic functions of geographic information systems. Traditional viewshed analysis algorithms are mainly carried out in a geospatial coordinate system, that create complexities when it is necessary to accurately express a perspective relationship. Moreover, it can only reduce the amount of calculation by using approximation methods, which result in decreased accuracy. We found that the error ratio of approximation algorithms is low but uneven and thereby highly likely to result in continuous area visibility errors. Despite the quality of DEM data having improved significantly in the last decade, traditional approximation algorithms have been unable to take full advantage of the improved data. Finding a fast and accurate algorithm has therefore become an urgent task. This study takes a new perspective to solve this problem by establishing a 'proximity-direction-elevation' (PDE) coordinate system and proposing a PDE spatial reference line (PDERL) algorithm within it. Many experiments prove that the accuracy of PDERL is the same as the R3 algorithm and its speed is about half that of the XDraw and reference plane algorithms, but much faster than R3. Due to the speed of PDERL, it is now possible and practical to do an accurate large-area DEM-based viewshed analysis.

Keywords Viewshed analysis · Digital elevation model · R3 algorithm · XDraw algorithm · Reference plane algorithm

Introduction

Digital elevation model (DEM) based viewshed analysis is one of the basic functions of many geographic information systems (GIS) (Qiang et al. 2019; Wang et al. 1996) and has broad applications. Teng et al. (1993) used viewshed analysis for topographic path planning, Winter-Livneh et al. (2012) and Bongers et al. (2012) used it for archaeological research, and Qiang et al. (2019) used it for visibility analysis of oceanic blue space. The study of viewshed analysis is based on the concept of line of sight (LOS), which refers to the line connecting an observer and a target (Franklin et al. 1994). When investigating visibility, it is necessary to know the standing positions of an observer, a target, and the intervening terrain. If any position along this line is blocked by terrain, the observer will not be able to observe the target; this phenomenon is referred to as invisibility while the converse is called visibility. Viewshed analysis refers to identifying ground areas that can be seen from a specified position within a certain range (Wang et al. 1996).

The spatial relation of viewshed analysis is very simple; however, the complexity lies in the number of calculations required. If no approximate methods are adopted, the number of calculations within traditional algorithms will be quite large. The R3 algorithm (Franklin et al. 1994) is the most direct and accurate viewshed analysis method that considers the impact of each grid line between the observer and target on the visibility. However, because the visibility of almost every target needs to be calculated individually, the computational burden is enormous, and the algorithmic complexity is $O(r^3)$, which limits its practical application, especially in large-area situations. The optimization of the computational requirements has always been a research hotspot in viewshed analysis. By using approximate methods, the R2 and XDraw algorithms proposed by Franklin et al. (1994), reduced the complexity of a viewshed analysis algorithm to $O(r^2)$; the reference plane algorithm proposed by Wang et al. (1996), pushed

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s12145-020-00545-7.

✉ Chuanjun Wu
blct_w@foxmail.com

[1] Institute of Geospatial Information, PLA Strategic Support Force Information Engineering University, Zhengzhou, China

the speed of the single-thread viewshed analysis algorithm to its peak at the time. On the basis of these traditional foundation methods and other similar studies, some scholars went on to improve the algorithm speed by reducing the computation on a single LOS (Liu et al. 2011; Ying 2005), optimizing data exchange (Fishman et al. 2009; Haverkort et al. 2009), adopting cluster computing (Llobera et al. 2010; Song et al. 2016), or using graphics processing unit (GPU) hardware (Cauchi-Saunders and Lewis 2015; Zhao et al. 2013). Research on improving viewshed analysis algorithm speed remains an important area of study (Dou et al. 2018; Dou et al. 2019; Li 2018).

Over the years, there have been numerous achievements in substantially increasing the speed but not the accuracy of viewshed analysis. Historically, most researchers believed that compared to the acquisition error of the DEM data itself, the calculation errors of fast approximation algorithms were relatively small, and their influence could be ignored. However, this study challenges that view. With the development of sensor technology, the resolution and precision of DEM products has been increasing, and the acquisition error of DEM data has been significantly reduced, especially in the last decade. For example, the AW3D enhanced product released by NTT RESTEC in 2014 had a resolution of 0.5 m, and the NEXTMAP ONE™ product released by Intermap in 2016 had an official elevation accuracy of 1 m (Li et al. 2018); this trend will improve with time. Traditional approximate fast algorithms cannot take advantage of the precision of modern DEM products. Furthermore, in addition to the error rate, we should also be concerned about error-point clustering. Although the algorithm has a low error rate if the error points are clustered together it will substantially impact the results. In short, high-quality viewshed analysis not only requires higher algorithm speed, but also cannot ignore the loss of accuracy. While the XDraw and reference plane algorithms are still the two most popular algorithms for the current parallel research (e.g. Cauchi-Saunders and Lewis 2015; Dou and Li 2018; Dou et al. 2018; Dou et al. 2019; Li 2018; Song et al. 2016), their calculation errors have the greatest impact on viewshed algorithms, and most of the improvements still focus on speed while ignoring accuracy. XDraw is the least accurate algorithm among the traditional approximation methods (Kaučič and Žalik 2002), and reference plane has the same accuracy problem because of their similar perspective relationship. This study finds that although the error rate of these traditional approximation algorithms is not high; the distribution of error points is uneven. In our experiment, more than half of the cases had at least one visibility error in a contiguous area, except over plain terrain, that would seriously affect the validity of the results. Such consequences are sometimes unacceptable; for example, in military applications, where a strategy made based on an unreliable calculation could lead to unintended and deadly consequences.

Because of the difficulty of accurate perspective calculation in geographic space, and the stereotype about the low accuracy of traditional DEM, only a few studies have investigated the issue of accuracy problem of viewshed analysis. Izraelevitz (2003) improved the accuracy of XDraw by backtracking along the LOS. The accuracy of this, however, depends on the number of backtracking intersections, thus making it more like a combination of XDraw and R3. Zhi et al. (2011) proposed that the reference plane algorithm was flawed for topographic calculation in areas such as peaks and valleys, and they proposed a solution to this problem by recording the historic minimum visible elevation of each target point and comparing to the LOS height. Although its accuracy improved by 56.91%, the computation time increased by 9.34 times (Zhu et al. 2019). Zhu et al. (2019) proposed an algorithm called HiXDraw to limit the chunk distortion of XDraw by finding the most related contributing points of the target. According to their research, this method achieved a 34.96% improvement in accuracy with 4.4 times the computation time, when compared with XDraw. Their research reduced the error rate of traditional approximation algorithms, but did not eliminate it, and it was achieved at a great cost to efficiency. The studies above show that the improvement of precision is of utmost importance, especially with respect to the problem of error point aggregation. However, due to the design of the algorithm itself, it is difficult for the traditional approximation algorithms to achieve advantages in both accuracy and speed.

In addition to the development of traditional algorithms, some new solutions have emerged. Wang et al. (2015) and Fang et al. (2011) used models to replace DEMs as the basic data for viewshed analysis, and integrated rendering and calculation into a GPU. In particular, the study by Wang et al. (2015) enhanced both algorithm speed and accuracy, presenting certain advantages over traditional DEM-based algorithms. However, this does not mean that viewshed analysis based on DEM is no longer valuable. Generally, the main function of viewshed analysis is not only to mark areas visually, but also to provide basic information for other analysis algorithms. In most cases, the results do not need to be displayed, and the overhead associated with model construction, analysis, and rendering are not necessary. To reduce the pressure of computer 3D visualization for large scenes, it is typical for many systems to simplify large-scale models through levels of detail (LOD) technology, model simplification algorithms, tile block loading, and other technologies (Li et al. 2017; Wu 2019). However, model-based viewshed analysis algorithms cannot apply the above technologies and maintain the accuracy of wide-range view calculations. The process of constructing accurate models from DEM data and then rendering them in wide-range calculations is expensive, and the overhead costs of the above steps have not been reported in the relevant literature. At present, DEMs are still the main method for terrain expression. When processing large-area viewshed analysis, the DEM-based algorithm remains the simplest and most direct method.

It is critical to reduce viewshed analysis errors while maintaining the algorithm speed. In contrast to the viewshed analysis studies in recent years which mainly focus on the parallelization of traditional fast algorithms, this study focuses on finding an accurate-fast algorithm that can replace the traditional approximate-fast algorithm. The three-dimensional geographic space, in which traditional basic non-parallel algorithms are always derived (e.g. Franklin et al. 1994; Izraelevitz 2003; Qiang et al. 2019; Wu 2001; Zhi et al. 2011; Zhu et al. 2019), leads to very concise position descriptions; however, perspective problems like viewshed analysis are complex and therefore the computational load can only be reduced by approximate calculation. If we use a coordinate system suitable for perspective, we can obtain an algorithm with both speed and accuracy. According to this principle, a coordinate system of 'proximity-direction-elevation' (PDE coordinate system) is established in this study. Its advantage is that the spatial linear relationship of lines in geographic space can be transformed into the linear relationship of direction, elevation, and proximity, which lays a foundation for fast and accurate viewshed analysis. Furthermore, a PDE spatial reference line algorithm (PDERL) is proposed based on the above coordinate system. After numerous experiments, we can prove that the speed of this algorithm is close to that of XDraw and reference plane algorithms and can therefore meet the speed requirements of practical applications. Moreover, this algorithm is precise, and its results are the same as the R3 algorithm, thus solving the problem of error point aggregation in the traditional approximate algorithms. The algorithm proposed in this study can alleviate the long-standing trade-off between speed and accuracy of DEM-based viewshed analysis algorithms.

Methodology

Materials

The experimental materials in this study consists of three DEMs from ASTER GDEM with 30 m grid spacing, in the TIFF format, whose serial numbers are N34E114, N41E119, and N28E097. The locations and topographic features of the three DEMs are shown in Table 1, and the height images of the DEM data are shown in Fig. 1. The three DEMs have a total area of approximately 36,000 km^2, and are broadly representative—covering typical landforms, including plains, hilly areas, and mountains—to ensure that the analysis results are universal. The algorithm in this study and other comparison algorithms are used to compare viewshed analyses based on these materials.

Fundamental basis

The visibility of the target is determined by the perspective relationship between the target and an obstruction. If there is an occlusion on the terrain, the following three conditions must be met simultaneously.

(1) The obstruction is in front of the target.
(2) The obstruction and the target are in the same direction from the observer.
(3) The elevation angle of the top LOS of the obstruction is greater than the top LOS of the target.

In this study, the PDE coordinate system is proposed to directly quantify the three conditions above. The description of the object in this space is transformed from the traditional geometric position coordinates into three conceptual dimensional coordinates relative to the observer: proximity (p), direction (d), and elevation (e). The PDE coordinate system is a right-handed coordinate system with p,d, and e as the X-axis, Y-axis, and Z-axis, respectively. If the two targets are in the same direction (d is the same) and the larger the value of e, the greater the elevation angle of the target LOS. Moreover, and the higher the value of p, the closer the target to the observer.

A DEM grid point is a spatial point with definite coordinates and elevation. Assuming that the grid line between the two points adjacent to the coordinates on the DEM can express the corresponding terrain, the continuous polyline formed by the DEM points of a row/column can express the terrain of the corresponding geographic space of the row/column, which is referred to as the latitude/longitude terrain line in this study. By definition of the spatial characteristics of PDE, any point located between and colinear with two points in geographic space is still located between and colinear with them in PDE space, so the polyline formed after terrain line is converted to PDE space can also represent any point on the original terrain line. When carefully dividing the calculation

Table 1 Overview of experimental materials (DEMs)

DEM	Location	Grid Spacing	Area	Terrain Type
N34E114	E114°–115°, N34°–35°	30 m	North China plain	Plain
N41E119	E119°–120°, N41°–42°	30 m	Northeast China	Hilly
N28E097	E97°–98°, N28°–29°	30 m	Sichuan, China	Mountains

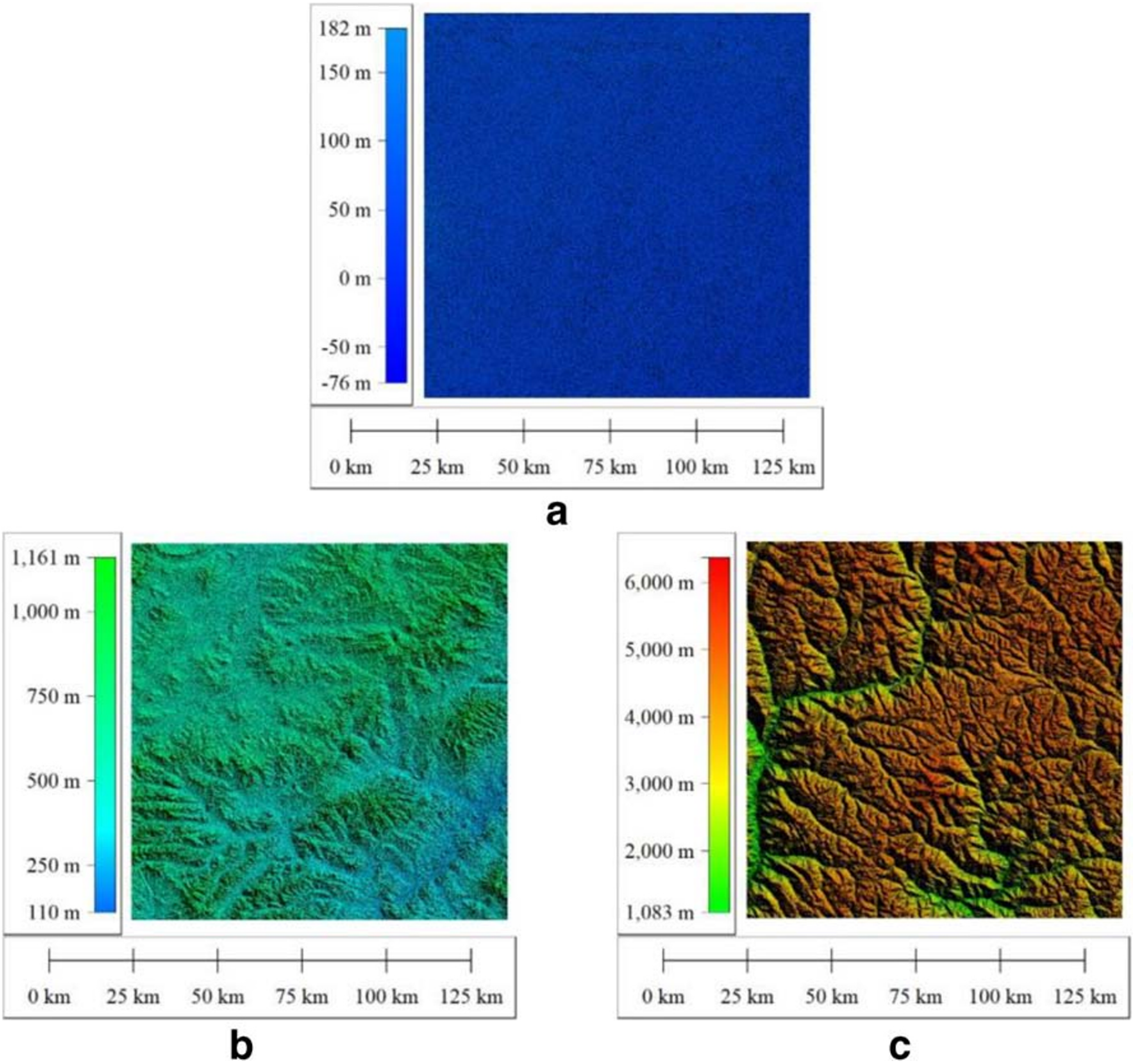


Fig. 1 The height images of the materials (DEMs). (**a**) N34E114, (**b**) N41E119, and (**c**) N28E097. They are plain terrain, hilly terrain, and mountainous terrain, respectively. As is shown, the topographic features of the three representative DEMs are quite different

regions, any terrain line converted to its corresponding PDE space will be in a plane with a constant proximity value and will also be a continuous polyline. The viewshed problem on a DEM can be abstractly discussed as the occlusion of the terrain line close to the observer (with larger proximity) to the terrain line far away from the observer (with smaller proximity). A latitude/longitude terrain line has an independent effect on the occlusion of grid points, but a grid point can be seen by the observer only if it is not shielded by both. Therefore, based on the position of the observer, this study divides the latitude and longitude terrain lines in the calculation region into four regions, namely Regions I, II, III, and IV, as shown in Fig. 2. Then each region is converted into the corresponding PDE space for calculation, and the visibility of any grid point will be judged by the latitude (Regions I and II) and longitude (Regions III and IV) terrain lines. The result can be obtained by combining the two. As this algorithm is based on the PDE coordinate system, this study introduces the definition and properties of the PDE coordinate system first, and then provides the derivation of PDERL in detail.

Definition of PDE coordinate system

The PDE coordinate system is defined based on a topocentric coordinate system. We define the topocentric coordinate system for each of the four regions. The topocentric coordinate system for Region I is defined as follows: take the point of view of the observer as the origin, the north direction of the DEM vertical grid line as the Y-axis, the east direction of the DEM horizontal grid line as the X-axis, and the positive direction of a normal vector on the reference ellipsoid of the observer as the Z-axis to establish the right-hand spatial rectangular coordinate system. While taking the Z-axis of the topocentric coordinate system in Region I as the axis of rotation, we can rotate the coordinate system clockwise to 180°, 270°, and 90°, to obtain other topocentric coordinate systems

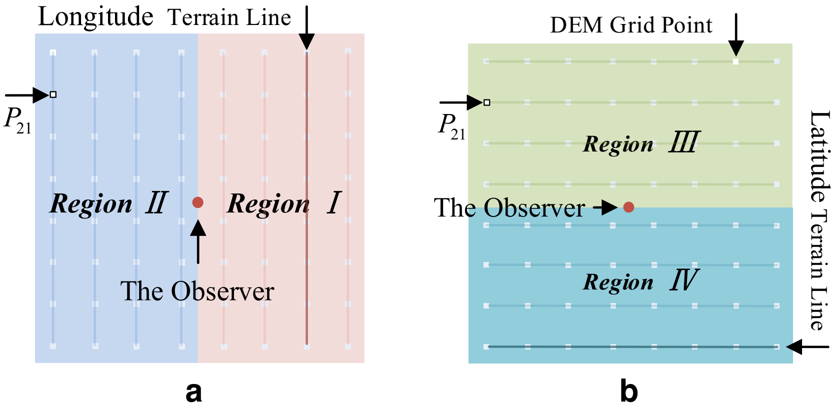


Fig. 2 The method of region division. (**a**) Division of Region I and Region II; (**b**) Division of Region III and Region IV. P_{21} in (**a**) is the same point as P_{21} in (**b**). The algorithm will check the influence of the longitude terrain lines and latitude terrain lines on the visibility of P_{21} in (a) and (b), respectively, and it will only be visible from the observer when both results are visible

in Region II, Region III, and Region IV, respectively. The geographic location of grid nodes in DEM files is generally recorded by the coordinates of longitude and latitude or a specific kind of map projection coordinates (such as Universal Transverse Mercator (UTM) projection), which record positions in space independent of the observer. When the coordinates of grid nodes are converted into the above topocentric coordinate system, they will be in relation to the relative spatial position of the observer.

This study defines three concepts—proximity, direction, and elevation—based on the topocentric coordinate system and uses them to quantify the three conditions of occlusion. The space constituted by the quantized value of these three concepts is defined as PDE space. This study introduces the definition of PDE space using point A as an example in Fig. 3a, and the same point transformed to PDE space as shown in Fig. 3b. We defined four topocentric coordinate systems that can be overlapped by the rotation described above. That said, the steps to convert coordinates from any topocentric coordinate system to PDE space coordinate system are the same. The definition below in this section will refer to the topocentric coordinate systems in Region I as an example. It should be emphasized that all coordinates (x, y and z) in the definitions of proximity, direction, and elevation below are the values in the topocentric coordinate system, not the coordinates of latitude and longitude or the map projection coordinates recorded by DEM itself.

The proximity p is defined as the reciprocal of the X-axis value of the investigation point in the topocentric coordinate system, and the definition is expressed as formula (1). The domain of p is$(-\infty, 0) \cup (0, +\infty)$. When two objects are in the same direction, the object with a larger proximity value (i.e. which is closer to the observer) may block the object with a smaller proximity value (i.e. which is farther away from the observer). This quantifies the occlusion condition (1).

$$p = 1/x \tag{1}$$

The direction d is defined as the tangent value of direction angle α in the topocentric coordinate system of the investigated point, and is defined by formula (2). The domain of d is$(-\infty, +\infty)$. When the d values of the two points are different, they are in different directions from the observer's perspective. This quantifies the occlusion condition (2).

$$d = \tan(\alpha) = yp \tag{2}$$

The elevation e is the product of the relative elevation z and proximity p of the observation point and is defined by formula

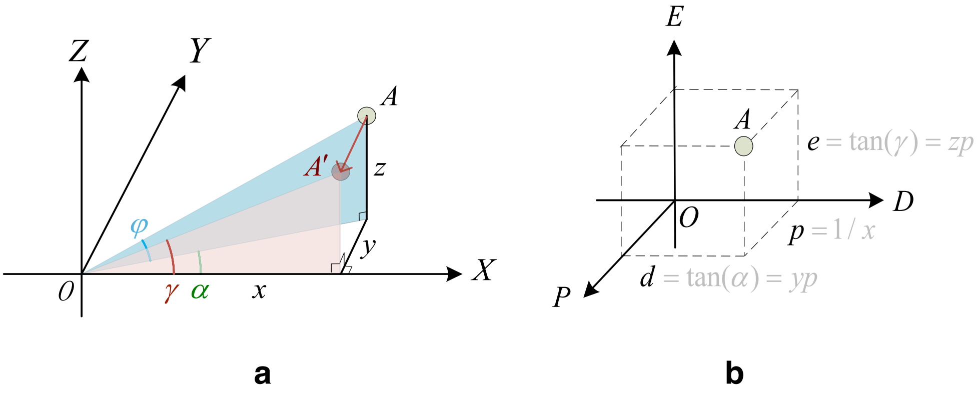


Fig. 3 Point A in different coordinate systems. (**a**) Point A in the topocentric coordinate system; (**b**) Point A in the PDE coordinate system. The coordinates of point A in topocentric coordinate system are (x, y, z); the LOS horizontal angle of A is α, and the elevation Angle of A is φ; the $A^{'}(x, 0, z)$ is obtained by shifting A along the Y-axis

(3). The domain of e is$(-\infty, +\infty)$. As in Fig. 3a, the meaning of e in geographic space is the tangent value of elevation γ, after point A is shifted along the Y-axis toA'. The slope (s) is the tangent value of elevation φ in the topocentric coordinate system; the higher the value of s, the higher the elevation angle of the LOS. Meanwhile, e and s conform to the relation formula (4). Therefore, when d is the same, the larger the value of e, the larger will be the value of s in the LOS. This quantifies the occlusion condition (3).

$$e = \tan(\gamma) = zp \tag{3}$$

$$e = s\sqrt{d^2 + 1} \tag{4}$$

The PDE coordinate system is defined as follows: take p as the X-axis, d as the Y-axis, and e as the Z-axis in a right-hand coordinate system. Combined with some mathematical principles, it is easy to prove that for any straight line in the topocentric coordinate system that is not in the Z-O-Y plane, when converted to the PDE space, the p, d, and e of the line are linearly related to each other, indicating that it remains a straight line in PDE space. This property lays a theoretical foundation for fast accurate calculation of the viewshed algorithm in this study.

The PDE spatial reference line algorithm

The essence of viewshed analysis in PDE space

The transformation from the topocentric coordinate system to PDE space coordinate system can be visualized as a complex perspective projection with an observer transformed into an orthographic projection where the observer is set at an infinity distant point on the P-axis and is looking in the opposite direction of the P-axis. When a terrain line has been converted to PDE space, it will be a perspective drawing of the terrain, and we call it the investigation line. When calculating the occlusion of an investigation line with a larger p to an investigation line with a smaller p, the former may block the later. The investigation line with a larger p is a reference when calculating the occlusion of any line with a smaller p, so we can call the former a reference line to the latter. Like the characteristic of orthogonal projection, we can judge visibility by the method of "overlay" of the investigation lines.

Taking Region I as an example, we convert the first terrain line C_1 to PDE space to form the initial reference lineL_R, as shown in Fig. 4, and convert the second terrain line C_2 to PDE space to constitute investigation lineL_2. Next, we shift L_R along the proximity axis to the plane where L_2 is, and divide the terrain line into several sections based on the intersection points between L_R andL_2. In fact, at the intersection of the two terrain lines, the visibility state of C_2 changes. The sections of the investigation line with a lower e will be covered and the grid points located on these sections will also be covered. Finally, we take the sections with higher e of L_R and L_2 to form a new terrain line L'_R, which in turn will be used as the reference line when calculating the terrain line C_3 in the next steps. Following these steps, we started from the first two terrain lines closest to the observer in one region and calculated the terrain line one by one according to the above steps until it was completed. The calculation steps of one region are shown in Fig. 5.

Expression of terrain line in PDE space

A terrain line is a spatial polyline constructed by a number of continuous DEM grid points. According to PDE spatial properties, for any segment of a terrain line there must be appropriate constants a and b that make the line segment conform to

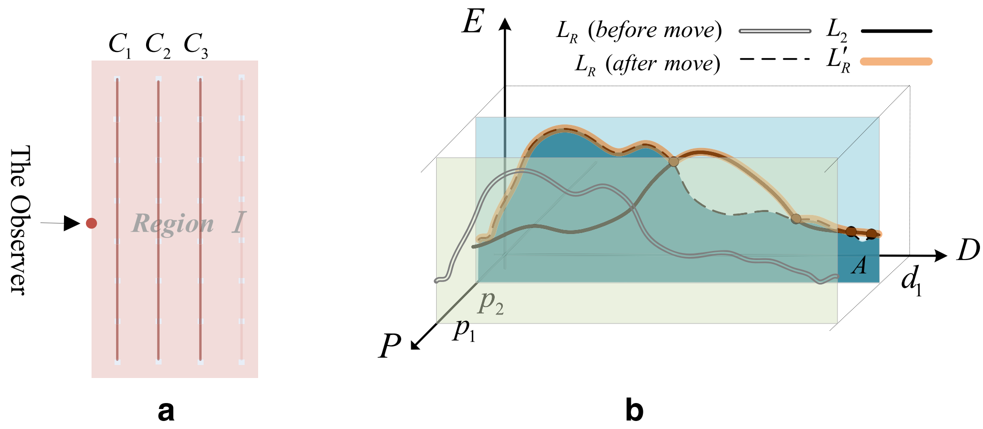
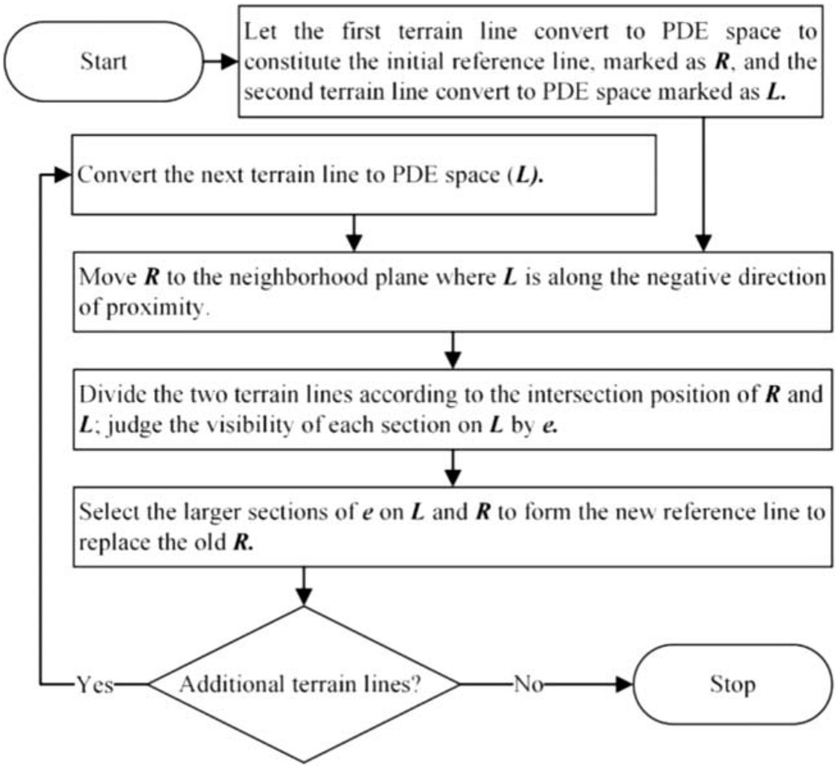


Fig. 4 Conversion of terrain line into the PDE space and updating of reference line. (a) The terrain line in geometric space. (b) Formation and updating of reference line in PDE space. During a reference line update in (b), the double line and the dashed line are the same reference line, and the latter is obtained from the former by shifting along the negative direction of the P-axis. The dark cyan surface A in (b) is the part of surface $p = p_2$ where elevation (e) is smaller than polyline L_R. It represents the part of area where L_R is capable of shielding the other investigation lines with a smaller proximity (e.g. L_2)

Fig. 5 Steps in each region



Let the first terrain line convert to PDE space to constitute the initial reference line, marked as ***R***, and the second terrain line convert to PDE space marked as ***L***.

Convert the next terrain line to PDE space (***L***).

Move ***R*** to the neighborhood plane where ***L*** is along the negative direction of proximity.

Divide the two terrain lines according to the intersection position of ***R*** and ***L***; judge the visibility of each section on ***L*** by ***e***.

Select the larger sections of ***e*** on ***L*** and ***R*** to form the new reference line to replace the old ***R***.

formula (5) in PDE space (see Supplementary Material 1 for the derivation). As $p = 1/x$ on each terrain line is constant, the terrain line must be a continuous polyline located in a PDE plane with constant proximity after being converted into PDE space. In this study, the line segment converted to PDE space is expressed as $L(a, b| d_l, d_r)$, where d_l and d_r are the left and right boundaries of the defined domain in the direction of the line segment and a, b are the parameters of L. If the topocentric coordinates of two adjacent grid points are known to be $A(x, y_A, z_A)$ and $B(x, y_B, z_B)$, and u is marked as the reciprocal of the grid width of two adjacent grid points, then the value of constant a can be expressed as formula (6) (see Supplementary Material 2 for the derivation).

$$e = ad + b \tag{5}$$

$$a = u(z_B - z_A) \tag{6}$$

In fact, b is not necessary to describe a terrain line. By recording e of the first segment, a of each segment, and the ending d of each segment, we can express and record the whole terrain line in PDE space.

Discriminant of visibility

As shown in Fig. 6, the difference ΔE_C by e_C, e_F is defined as formula (7), corresponding to the vector in Fig. 6. If r_C is used to mark the visibility of point C, the value of 0 indicates invisibility, and the value of 1 indicates visibility; then, we can easily obtain the discriminant of visibility as formula (8).

Construction and update of reference line

The calculation of the visible segment on the investigation line and the update of the reference line are a cumulative process of ΔE. As shown in Fig. 7, the current investigation line is labelled L_C and the current reference line is labelled L_R which has been moved to the plane where L_C is located. The

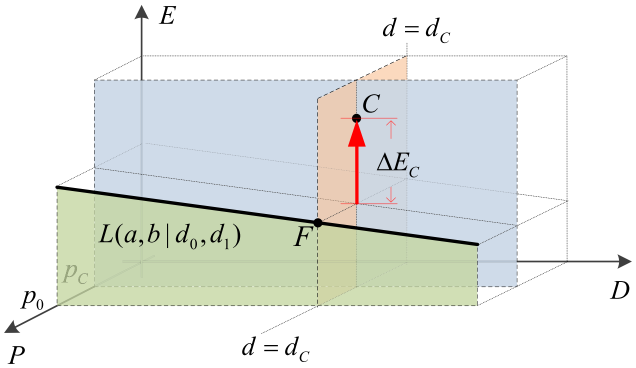


Fig. 6 The definition of ΔE_C. Parameters and definition domain of $L(a, b| d_0, d_1)$ are known, while proximity is p_0. The PDE coordinate of point C in PDE is(p_C, d_C, e_C), and$p_C < p_0$, $d_C \in [d_0, d_1]$. $L(a, b| d_0, d_1)$ intersects the PDE plane $d = d_C$ at point F, and the intersection coordinate is (p_0, d_C, e_F)

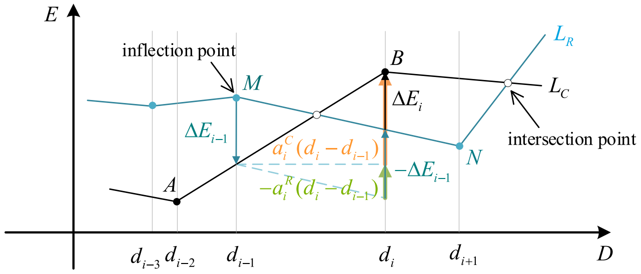


Fig. 7 Relationship between ΔE_i and ΔE_{i-1}. The figure shows a PDE plane where L_C is located, and L_R has been translated from a plane with a bigger proximity to this plane. L_R is the reference line to L_C. The values of parameter a of AB and MN are a_i^C and a_i^R. It is evident from the figure that $\Delta E_i - \Delta E_{i-1} = \left(a_i^C - a_i^R\right)\left(d_i - d_{i-1}\right)$, which is the theoretical basis of the cumulative formula for ΔE

direction values of all the inflection points of L_R and L_C are taken as the segmentation points with which the two terrain lines are divided into several segments. The direction at the end of segment i is labelled d_i, and the values of parameter a on L_R and L_C in this segment are a_i^C and a_i^R. The value of ΔE at the end of this segment is ΔE_i, while at the end of the previous segment it is ΔE_{i-1}. Due to the linear characteristics, it is easy to obtain formula (9), the cumulative formula for ΔE. If the end of a section is a grid point on the investigation line, the visibility of the grid point can be obtained by a combination of formulas (8) and (9). If ΔE_{i-1} and ΔE_i have opposite positive and negative values, then L_R and L_C have an intersection point in segment i, and the direction of the intersection point (d') can be solved by formula (10). By using formulas (7)–(10), we can follow the steps in Fig. 8 to calculate the visibility of one investigation line and update the current reference line.

When we follow the last step of Fig. 8, the identified intersection points (generally not grid points) were added to the new reference line as inflection points when we selected the sections with higher e to update the reference line. Thanks to the precise representation of the reference lines by these non-grid points, the accuracy of the PDERL algorithm results will be comparable to those of the R3 algorithm. The process of updating the reference line is the main computational process of PDERL, and the formulas (8)–(10) will be the most utilized in this process. We derived these formulas in the PDE coordinate system; however, it is not necessary for an independent step to convert DEM from its recorded coordinate system into the PDE coordinate system in the actual calculation. As all the above formulas can be converted into equivalent mathematical formulas expressed by geographic space coordinates through the definitions, when we use them for viewshed analysis, it already contains the coordinate transformation process.

$$\Delta E_C = e_C - e_F \tag{7}$$

$$r_C = \begin{cases} 0, & \Delta E_C < 0 \\ 1, & \Delta E_C \geq 0 \end{cases} \tag{8}$$

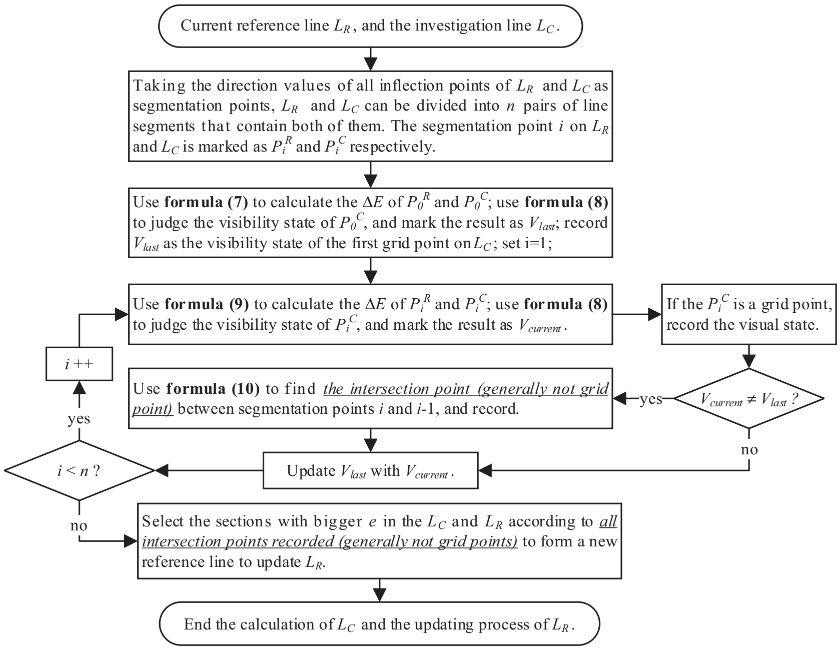
$$\Delta E_i = \Delta E_{i-1} + \left(a_i^C - a_i^R\right)\left(d_i - d_{i-1}\right) \tag{9}$$

$$d' = d_i - \Delta E_i / \left(a_i^C - a_i^R\right) \tag{10}$$

Consolidation viewshed results of the four regions

Grid points can be judged to be visible only when they are determined to be visible by the longitude terrain lines (in Region I/II) and the latitude terrain lines (in Region III/IV) simultaneously. In this study, a 0–1 matrix is used to record

Fig. 8 Steps of visibility calculation of one general investigation line


Current reference line L_R, and the investigation line L_C.
Taking the direction values of all inflection points of L_R and L_C as segmentation points, L_R and L_C can be divided into n pairs of line segments that contain both of them. The segmentation point i on L_R and L_C is marked as P_i^R and P_i^C respectively.
Use **formula (7)** to calculate the ΔE of P_0^R and P_0^C; use **formula (8)** to judge the visibility state of P_0^C, and mark the result as V_{last}; record V_{last} as the visibility state of the first grid point on L_C; set i=1;
Use **formula (9)** to calculate the ΔE of P_i^R and P_i^C; use **formula (8)** to judge the visibility state of P_i^C, and mark the result as $V_{current}$.
If the P_i^C is a grid point, record the visual state.
$V_{current} \neq V_{last}$?
yes
no
Use **formula (10)** to find the intersection point (generally not grid point) between segmentation points i and i-1, and record.
i ++
yes
$i < n$?
Update V_{last} with $V_{current}$.
no
Select the sections with bigger e in the L_C and L_R according to all intersection points recorded (generally not grid points) to form a new reference line to update L_R.
End the calculation of L_C and the updating process of L_R.

viewshed analysis results, which have the same dimensions as the viewshed DEM, and its default value is 0. The results of viewshed analysis for each grid point are stored in the corresponding element with the same row and column number. The value of 0 represents invisibility, and 1 represents visibility. The results of Regions I, II, III, and IV are recorded in the visibility matrix V_1, V_2, V_3, and V_4. The final visibility matrix is obtained using formula (11).

$$V = (V_1 + V_2)\cdot(V_3 + V_4). \tag{11}$$

Experiments and results

To verify the effectiveness of the algorithm, we designed four contrast experiments on PDERL, R3, XDraw, and reference plane algorithms to ascertain accuracy, aggregation degree of error points, computational complexity, and speed. As a classical viewshed analysis algorithm, R3 considers all grid lines in LOS as they affect the visibility. The results of R3 are taken as the standard because of their high accuracy and reliability and are used to evaluate the error points of the results of the other algorithms. This study implements all the compared algorithms with C# programming in a back-end web service, while developing a front-end interactive platform with Cesium. These algorithms run on a Mi computer 181,501-AB with a Windows 10 operating system.

Experiment 1: Accuracy

This experiment began at an altitude of 1 m above the ground. For every 25 m increase in height, the observation location and analysis radius were randomly selected (radius: 270–10,680 m). The experiments were conducted 50 times for each of three DEM blocks. For each experiment, PDERL, XDraw, reference plane, and R3 were used to calculate the same region, and the results of each algorithm were compared with the results of R3 to analyze the corresponding errors. The statistical results are shown as polynomial fitting curves of the experimental data in Fig. 9. By comparing the upper limit of error rate, average error rate, and median error rate of each algorithm, it can be observed that:

(1) The results of PDERL in all experiments are consistent with R3;
(2) The error of the XDraw/reference plane algorithm is the largest near the ground and decreases gradually with the increase in observation height;
(3) The XDraw/reference plane algorithms have a higher error rate near the ground in the flat terrain area than in the undulating terrain area, but it declines rapidly with the height from the ground.

Experiment 2: Aggregation of error points

The PDERL algorithm has no error points; accordingly, there is no possibility of aggregating error points as in most experimental results of XDraw and reference plane algorithms. For example, Fig. 10 shows the viewshed analysis results calculated by PDERL and XDraw algorithms in an experiment. The coordinate of the observation center is 119.8219° E, 41.7602° N, 5 m above the ground, and the observation radius is 1 km. As shown in the figure, there are some error point aggregations in the XDraw result while there are no error points in the PDERL result. In fact, the error rate of XDraw algorithm is not high in this experiment, only 3.1221%, but such a low error rate still cannot prevent the problem of many error points becoming aggregated.

To quantify the degree of aggregation of the error points in the results, this study defines the concept of an 'x-neighborhood area error (x-NE)' and an x-NE logical flag (x-NEF) corresponding to whether one result consists of at least one x-NE (1) or not (0). An x-NE means the area where 90% of the center grid point and the grid points in its x–grid distances around (as shown in Fig. 11) have the same kind of visible error. For the x-NEF, first select an algorithm to perform a viewshed analysis and then check whether the analysis result consists of at least one x-NE. The value of the x-NEF is 1 for

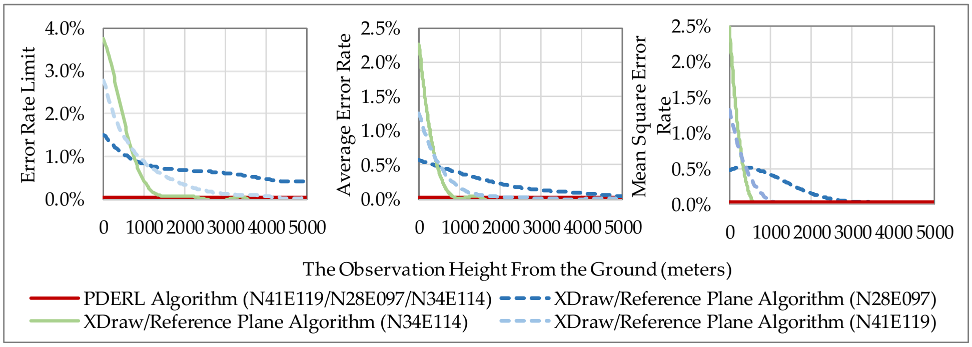


Fig. 9 Error rate statistics of the algorithms on three terrains. The left/middle/right subfigure is the error rate of limit/average/mean square of PDERL, XDraw and reference plane algorithms sorted by height above the ground. Since the XDraw and reference plane algorithms have the same error condition, they are drawn on the same line

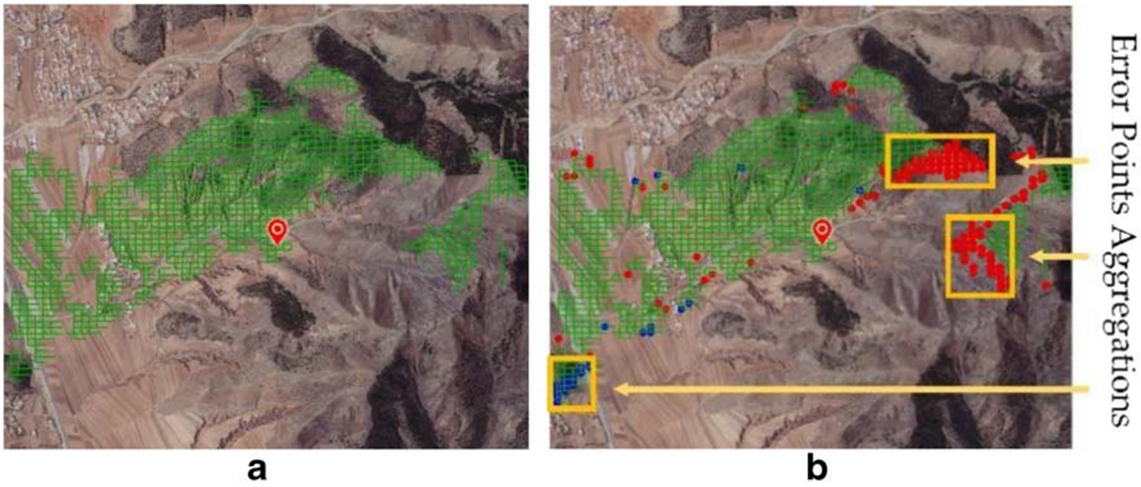


Fig. 10 Viewshed analysis results of different algorithms in the same region. (**a**) One Result of PDERL; (**b**) One Result of XDraw. The areas showing the grid are the correct visibility areas. The white dots are the cases where visibility is incorrectly judged as invisible, and the dark dots are the cases where invisibility is wrongly judged as visible. Obviously, there is significant aggregation of error points in the XDraw result

yes or 0 for no, instead of a continuous value, and its main function is to count how many results are affected by x-NEs by summing the x-NEFs. The DEM resolution tested in this study is 30 m, and the size of the area or grid points involved in various types of x-NEs are shown in Table 2.

In this experiment, heights from 2 to 52 m above the ground (at 2 m intervals) for three DEM sections (covering 120.5604 km^2 each) with a fixed radius (5.49 km) and randomly selected location were computed 500 times with R3, PDERL, and XDraw. This comparative experiment was carried out 12,500 times on each of the three DEMs and 1500 times for each height. Considering the large number of experiments undertaken, the results were representative to some extent. In this study, there was no error point in the results of PDERL, while there were many error points in the results of Xdraw. The x-NEFs of the results of Xdraw algorithm were counted as shown in Fig. 12 and Table 3 according to their height order and DEM file. As can be seen from the relevant figures and tables, (a) in mountainous (N28E097) and hilly (N41E119) areas, more than half of the calculation results have 1-NEs; (b) the larger the range of x, the lower the probability of x-NEs. Furthermore, in the calculation results of the mountainous area (N28E097) the occurrence frequency of 10-NE is still 0.15%, which means that in this experiment 0.15% of N28E097 calculation results experienced at least 396 consecutive grid-point errors, involving a continuous error area of up to 396,900 m^2.

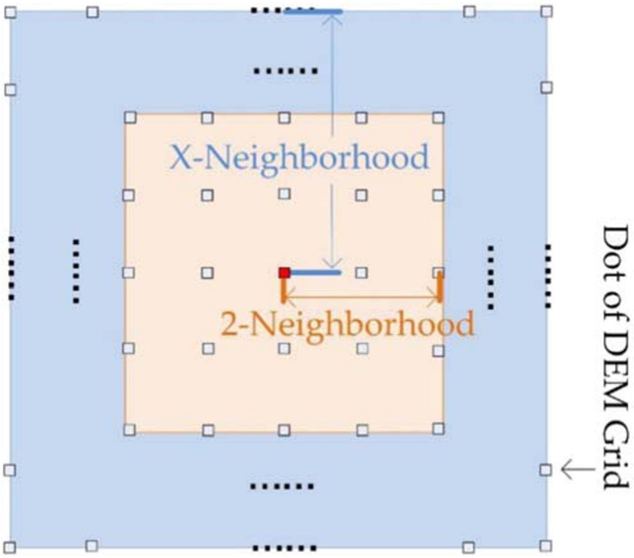


Fig. 11 Neighborhood of DEM grid points. X is the number of grids spacing of the radius

Experiment 3: Computational complexity

The PDERL and R3 algorithms are both exact algorithms; however, the complexity of R3 is $O(r^3)$ and the theoretical complexity of PDERL is the same as XDraw and reference plane, which is $O(r^2)$. Theoretically, PDERL should

Table 2 Area covered by each x-NE type and the number of grid points

The Type of x-NE	Total Grid Points	90% of Grid Points	Area (m^2)
1-NE	9	8	8100
2-NE	25	22	22,500
3-NE	49	44	44,100
4-NE	81	72	72,900
5-NE	121	108	108,900
6-NE	169	152	152,100
7-NE	225	202	202,500
8-NE	289	260	260,100
9-NE	361	324	324,900
10-NE	441	396	396,900

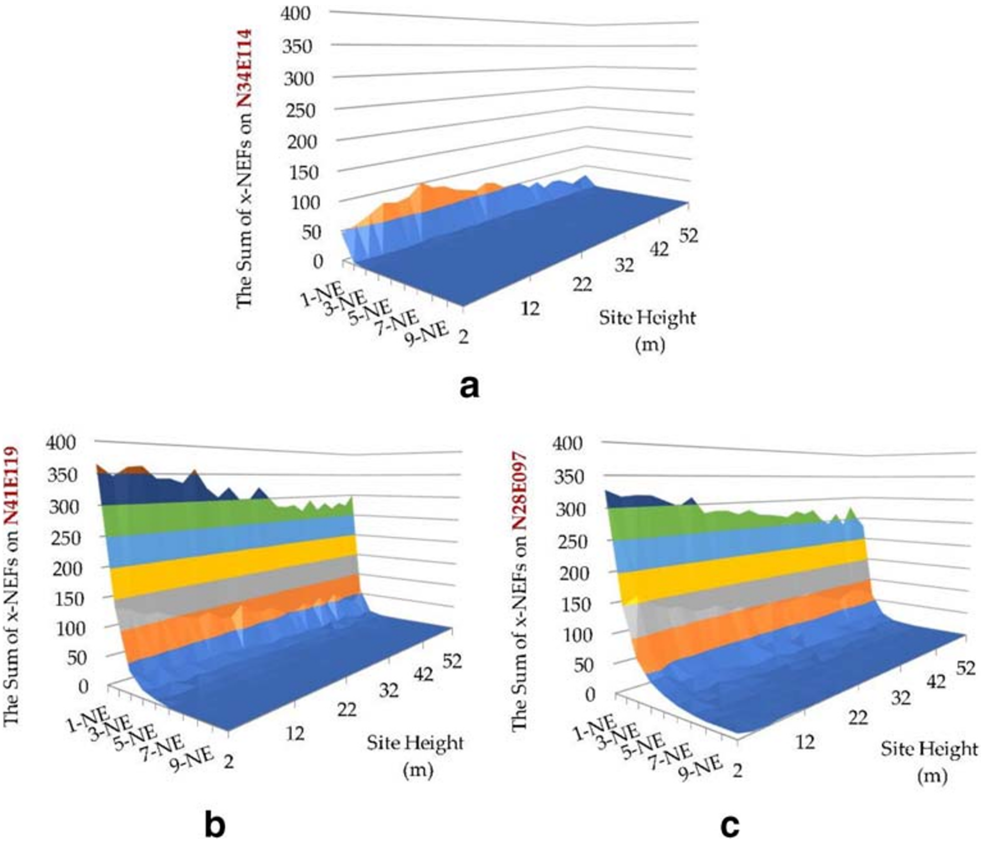


Fig. 12 The sum of x-NEFs on different DEMs and different site Heights. (**a**) The sum of x-NEFs on plain terrain (N34E114); (**b**) The sum of x-NEFs on hilly terrain (N41E119); (**c**) The sum of x-NEFs on mountainous terrain (N28E097). Site height is the height above the ground

be much faster than R3 and closer to XDraw/reference plane. To compare their computational complexities, an experiment was conducted on the DEM N41E119 at a height of 2 m from the ground, radius from 300 m to 4800 m (at 30 m intervals), computed 11 times with R3, PDERL, XDraw, and reference plane on randomly selected locations. In this study, the computation times of each algorithm was plotted statistically in order of radius size (Fig. 13). The experimental results confirm our assumption on the degree of computational complexity of the considered algorithms.

Experiment 4: Speed

To compare the speed of PDERL to that of XDraw, reference plane, and other recent improvements in accuracy of

Table 3 Distribution of x-NE rate of XDraw algorithm on three kinds of terrain

The Type of x-NE	N28E097 (Mountainous)	N41E119 (Hilly)	N34E114 (Plain)
1-NE	58.35%	64.63%	11.24%
2-NE	21.36%	15.87%	0.30%
3-NE	6.94%	3.07%	0.06%
4-NE	3.19%	0.98%	0.01%
5-NE	1.58%	0.30%	0.01%
6-NE	0.79%	0.06%	0.01%
7-NE	0.51%	0.02%	0.00%
8-NE	0.32%	0.00%	0.00%
9-NE	0.23%	0.00%	0.00%
10-NE	0.15%	0.00%	0.00%

Note that the distribution ratio is obtained by dividing the sum of x-NEFs on different DEM by the total number of experiments on the corresponding DEM, respectively. It is an indicator used to measure the probability that a kind of x-NE will affect the result of a viewshed analysis.

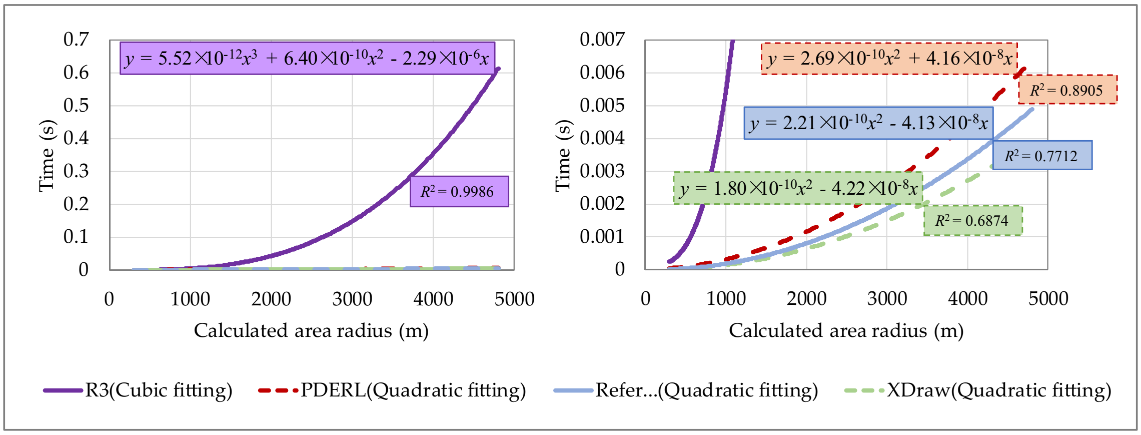


Fig. 13 Run times for the different algorithms (all data preparation times have been excepted). The formula of each line is the function of polynomial fitting, and R^2 is the residual sum of squares of the corresponding fitting. As is shown in the left subfigure, the time consumption of R3 is much larger than the other algorithms, with the complexity of $O(r^3)$. To clearly express the time costs of PDERL, XDraw and reference plane, the times from 0 to 0.007 s are plotted separately in the right subfigure, which clearly proves that the speeds of the three are close to each other with the same complexity of $O(r^2)$

algorithms, for example as in Zhu et al. (2019), this study conducted 6 experiments with each of the three algorithms, on the three DEMs, at heights from 1 m to 5000 m from the ground (in 1 m intervals), with a radius of 10.77 km, at randomly selected locations, totaling 15,000 comparison experiments. We used a concept called time consumption ratio (R), similar to Zhu's approach (Zhu et al. 2019), to evaluate the speed difference between PDERL, XDraw, and reference plane. Time consumption of PDERL, XDraw, and reference plane were defined as T_{PDERL}, T_{XDraw}, and T_{rp}, respectively. They were compared in an experiment through their time consumption ratio between XDraw (R_{XDraw}) and reference plane (R_{rp}) in n experiments with the same height as defined by formula (12).

$$\begin{cases} R_{XDraw} = (1/n)\sum(T_{PDERL}/T_{XDraw}) \\ R_{rp} = (1/n)\sum\left(T_{PDERL}/T_{rp}\right) \end{cases} \tag{12}$$

After fitting with a 6th degree polynomial, the results of R_{XDraw} and R_{rp} in the order of height in different terrains are shown in Fig. 14 below. The speed performance of the PDERL algorithm is not very stable, particularly at a low height, but the overall difference in speed is not large, a little faster than half the speed of XDraw, and a little slower than half the speed of reference plane. Although we did not implement the method in Zhu et al. (2019) we made a comparison based on their research. Their algorithm is 4.4 times as time-consuming as XDraw with limited accuracy improvement (Zhu et al. 2019), while PDERL is at most 2.25 times as time-consuming as XDraw with completely accurate results.

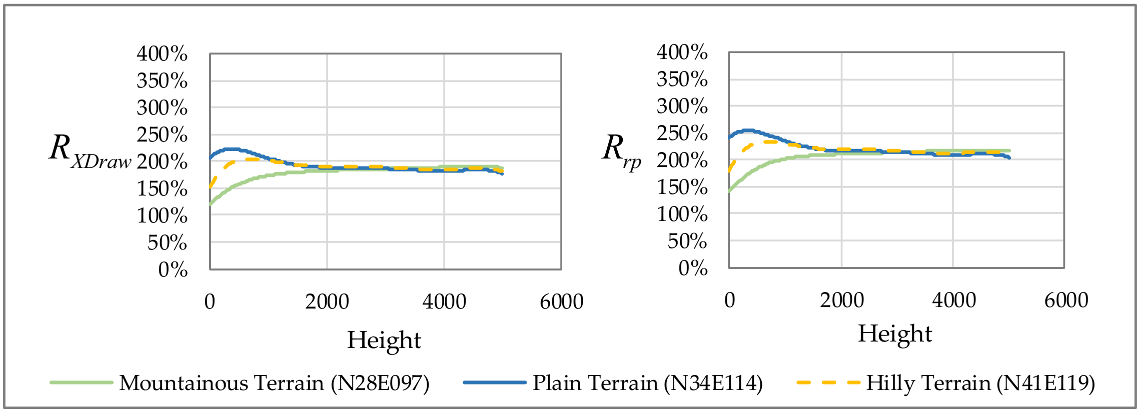


Fig. 14 The time consumption ratio of PDERL to XDraw and reference plane. The left/right subfigure shows the time consumption ratio of PDERL to XDraw and reference plane at different heights from the ground. While the time consumption of this experiment is relatively stable at high altitudes, it is approximately 185% of XDraw and 215% of the reference plane, and the maximum value of the experiment is not more than 225% of XDraw and 255% of reference plane

Discussion and conclusion

With the rapid improvement in accuracy of modern DEM products, the calculation error present in the traditional fast approximation algorithms has become the main source of error in viewshed analysis (Zhu et al. 2019). Traditional viewshed analysis algorithms are mainly carried out in the geospatial coordinate system, which encounters complexities and difficulties when required to accurately express the perspective relationship. Moreover, efficiency can only be improved by using approximation methods, which result in a decline in accuracy. According to the results of experiment 1, the problem of loss of accuracy is especially serious in mountainous/hilly regions and near ground level. Due to the design of the approximation methods, as previous researches have shown, accuracy improvements based on traditional approximation algorithms may reduce the error rate but will not eliminate it (e.g. Izraelevitz 2003; Zhi et al. 2011; Zhu et al. 2019).

The influence of accuracy on the results of the viewshed analysis is not limited to the error rate. According to experiment 2, although the ratio of error points in the traditional fast approximate algorithms is not high, many of their results show error point aggregation. This problem will lead to regional visibility misjudgment and it will have a great impact on applications with high precision requirements. The effect of error point aggregation is more severe than error rate.

Compared to the traditional viewshed analysis algorithms and their improved methods, PDERL has a great advantage in terms of accuracy. According to the results of experiments 1 and 2, PDERL is as precise an algorithm as R3, however, the former is much faster as can be seen from the results of experiment 3. PDERL can take full advantage of modern DEMs by avoiding calculation errors and is free of error point aggregations. Therefore, it is more suitable for applications with high precision requirements.

Benefitting from the speed of PDERL, we now can apply a large-area DEM-based precise viewshed analysis in practice. Although the speed of PDERL is not as fast as that of XDraw and the reference plane algorithm, as can be seen from the results of experiment 3 and 4, their computational complexities are comparable, and the speed is similar to those of the latter two. Furthermore, as shown in experiment 1 and 4, PDERL can achieve full accuracy under the premise of speed, which is much more efficient than previous precision improvements based on traditional approximation algorithms (Izraelevitz 2003; Zhi et al. 2011; Zhu et al. 2019), and hence, it provides superior results for many applications.

PDERL provides a new solution with excellent performance. Instead of the traditional geospatial coordinates, a newly established PDE space is proposed. The coordinate system of the PDE space can concisely express the perspective relationship with the observer. Furthermore, it can convert the spatial linear relationship of the lines in the geographic space into the linear relationship of direction, elevation, and proximity, and thus, lay the foundation for a fast and accurate viewshed analysis. In PDE space, instead of interpolation or any other approximation method, the occlusion relations of the terrain can easily and precisely propagate in the form of a linear piecewise function through the updating of the reference line from near to far. This is an inspiration for the study of other perspective problems in the geographic space and deserves further research.

Of course, PDERL is not perfect. Although it is extremely fast among the accurate algorithms, its speed is only about half that of XDraw and reference plane, and thus, it needs to be further improved. Taking advantage of modern parallelization methods to improve the speed of the PDERL algorithm in the future is essential. An interesting phenomenon was observed in experiment 4 indicating that the speed of the PDERL algorithm is not very stable, particularly at a low height because the intersection operation of the terrain line in the PDE space is required during the PDERL calculations. Moreover, different intersection points may affect the cumulative times of ΔE. Furthermore, the calculation processes of the visible and invisible parts of the terrain line are slightly different; and therefore, the algorithm speed is related to the terrain complexity, visibility rate, and many other factors. However, the overall speed difference among the PDERL, XDraw, and reference plane algorithms is not large.

In conclusion, our study provides a novel perspective in the use of a non-geographical coordinate system to research existing viewshed analysis problems. Although it needs further improvement in terms of speed, PDERL successfully makes the large-area DEM-based precise calculation of viewshed analysis practical and avoids error point aggregation. Thus, it effectively alleviates the long-standing contradiction between speed and accuracy in this field.

Acknowledgements We would like to thank Editage (www.editage.cn) for English language editing, Geospatial Data Cloud for providing open DEM Data, Mr. Jian Yang, Mr. Jiangpeng Tian and Mr. Yongshan Ren for their helpful advice, and anonymous editors and reviewers for their valuable comments on this article.

Availability of material and code The experimental materials and back-end code that support the findings of this study are available in Figshare with the private link https://figshare.com/s/4e998f3f5ae766331bfe. Front-end interactive platform is not made public because it is not essential, easy to replace, and needs complex deployment.

Funding This research received no external funding.

Compliance with ethical standards

Conflicts of interest/competing interests No potential conflict of interest was reported by the authors.

References

Bongers J, Arkush E, Harrower M (2012) Landscapes of death: GIS-based analyses of chullpas in the western Lake Titicaca basin. J Archaeol Sci 39:1687–1693. https://doi.org/10.1016/j.jas.2011.11.018

Cauchi-Saunders AJ, Lewis IJ (2015) GPU enabled XDraw viewshed analysis. J Parallel Distrib Comput 84:87–93. https://doi.org/10.1016/j.jpdc.2015.07.001

Dou W, Li Y (2018) A fault-tolerant computing method for Xdraw parallel algorithm. J Supercomput 74:2776–2800. https://doi.org/10.1007/s11227-018-2321-x

Dou W, Li Y, Wang Y (2018) A fine-granularity scheduling algorithm for parallel XDraw viewshed analysis. Earth Sci Inf 11:433–447. https://doi.org/10.1007/s12145-018-0339-5

Dou W, Li Y, Wang Y (2019) An equal-area triangulated partition method for parallel Xdraw viewshed analysis. Concurrency Computat Pract Exper 31:1–14. https://doi.org/10.1002/cpe.5216

Fang C, Yang C, Chen Z, Yao X, Guo H (2011) Parallel algorithm for viewshed analysis on a modern GPU. Int J Digit Earth 4:471–486. https://doi.org/10.1080/17538947.2011.555565

Fishman J, Haverkort H, Toma L (2009) Improved Visibility Computation on Massive Grid Terrains. In: Agrawal D, Aref WG et al (eds) Proceedings of the 17th ACM Conference on advances in geographic information systems, 4–6, 2009. ACM, Seattle, pp 121–130

Franklin WR, Ray CK, Mehta S (1994) Geometric algorithms for siting of air defense missile batteries. Research project for Battle order 2756, Columbus

Haverkort H, Toma L, Zhuang Y (2009) Computing visibility on terrains in external memory. ACM J Exp Algor 13(1.5):1–1.5:23. https://doi.org/10.1145/1412228.1412233

Izraelevitz D (2003) A fast algorithm for approximate viewshed computation. Photogramm Eng Remote Sens 69:767–774. https://doi.org/10.14358/PERS.69.7.767

Kaučič B, Žalik B (2002) Comparison of Viewshed algorithms on regular spaced points. In: SCCG '02: Proceedings of the 18th Spring Conference on Computer Graphics, Association for Computing Machinery, New York, NY, USA, pp 177–183

Li Y (2018) Design and implementation of a parallel Viewshed analysis algorithm based on XDraw. Dissertation, Nanjing Normal University, Nanjing, China

Li X, Deng F, Li X, Wu S (2017) A visualization plan for large-scale 3D models with oblique photogrammetry technology. Bull Surv Mapp 2017:39–43

Li Z, Li R, Ding D, Wang H (2018) Research Progress of Global High Resolution Digital Elevation Models. Geomatics and Information Science of Wuhan University 43:1927–1942 doi:https://doi.org/10.13203/j.whugis20180295

Liu L, Zhang L, Zhang L, Zhang X, Yang L (2011) An improved line-of-sight method for visibility analysis in3D complex landscapes. Scientia Sinica Informationis 41:675–685

Llobera M, Wheatley D, Steele J, Cox S, Parchment O (2010) Calculating the inherent visual structure of a landscape (total viewshed) using highthroughput computing. In: Niccolucci F, Hermon S (eds) Beyond the artifact: digital interpretation of the past, CAA 04. Budapest, Archaeolingua, pp 146–151

Qiang Y, Shen S, Chen Q (2019) Visibility analysis of oceanic blue space using digital elevation models. Landsc Urban Plan 181:92–102. https://doi.org/10.1016/j.landurbplan.2018.09.019

Song X-D, Tang G-A, Liu X-J, Dou W-F, Li F-Y (2016) Parallel viewshed analysis on a PC cluster system using triple-based irregular partition scheme. Earth Sci Inf 9:511–523. https://doi.org/10.1007/s12145-016-0263-5

Teng YA, DeMenthon D, Davis LS (1993) Stealth terrain navigation. IEEE trans Syst man Cybern 23:96–110. https://doi.org/10.1109/21.214770

Wang J, Robinson GJ, White K (1996) A fast solution to local Viewshed computation using grid-based digital elevation models. Photogramm Eng Remote Sens 62:1157–1164

Wang F, Wang G, Pan D, Liu Y, Yang L, Wang H (2015) A parallel algorithm for viewshed analysis in three-dimensional digital earth. Comput Geosci 75:57–65. https://doi.org/10.1016/j.cageo.2014.10.012

Winter-Livneh R, Svoray T, Gilead I (2012) Secondary burial cemeteries, visibility and land tenure: a view from the southern Levant chalcolithic period. J Anthropol Archaeol 31:423–438. https://doi.org/10.1016/j.jaa.2012.03.002

Wu Y (2001) An algorithm for computing viewsheds based on reference planes. J Geomat 2001:19–21. https://doi.org/10.14188/j.2095-6045.2001.01.007

Wu W (2019) Research on large scene natural environment simulation based on virtual reality technology. Dissertation, Jiling University, Jiling, China

Ying S (2005) Key techniques and applications of Spqtial visibility analysis. Dissertation, Wuhan University, Wuhan, China

Zhao Y, Padmanabhan A, Wang S (2013) A parallel computing approach to viewshed analysis of large terrain data using graphics processing units. Int J Geogr Inf Sci 27:363–384. https://doi.org/10.1080/13658816.2012.692372

Zhi Y, Wu L, Sui Z, Cai H (2011) An improved algorithm for computing viewshed based on reference planes. Paper presented at the 19th International Conference on Geoinformatics, Shanghai, pp 24–26 June 2011

Zhu G, Li J, Wu J, Ma M, Wang L, Jing N (2019) HiXDraw: an improved XDraw algorithm free of chunk distortion. ISPRS Int Geo-Inf 8:153–169. https://doi.org/10.3390/ijgi8030153