A Fast Solution to Local Viewshed Computation Using Grid-Based Digital Elevation Models

Jianjun Wang, Gary J. Robinson, and Kevin White

Abstract

A new algorithm for generating viewsheds from grid-based digital elevation models is presented. It uses a combination of sightline analysis and the relationships between the local surfaces at the source and destination points. Depending on the nature of the terrain surface, significant savings in computation time over existing sightline only based algorithms are made because a substantial proportion of the destination points can be determined as invisible from the source point without the need for sightlines. The new algorithm also produces more reliable results by better consideration of points at or near the viewshed horizon.

Introduction

A viewshed is the area on the ground that is visible from a specified location. Viewsheds are employed in a number of applications. For example, fire monitoring towers in forests need to be situated in locations with large viewsheds, whereas the optimum path for moving troops should have the smallest overall viewshed.

Viewshed analysis using digital elevation models (DEMs) has become one of the standard functions of geographical information systems (GIS). Of the two main types of DEM used in viewshed analysis — grid-based DEMs and triangulated irregular networks (TINs) — the former is more widely used because its data structure is simpler and therefore more convenient for computers to manipulate. However, an unfortunate drawback of gridded DEMs is that their spacing is often selected as a compromise between the ability of the DEM to resolve small features (within appropriate scale limits) while maintaining a manageable number of points. Clearly, an undersampled terrain surface may in reality contain peaks and depressions that are higher or lower than those predicted, e.g., by bilinear interpolation from the DEM, and this may affect the results of the viewshed computation. However, we ignore this potential problem in the subsequent discussion and assume that we are dealing with optimally sampled grid-based DEMs exclusively in terms of viewshed generation.

A grid-based DEM (Figure 1) can be defined as a matrix of M rows and N columns: i.e., DEM=$\{z_{i,j}\}$, $(i=1,2,...,M;$ $j=1,2,...,N)$, where $z_{i,j}$ represents the elevation at the matrix position (i,j) and where the points of the DEM are equally spaced along the X and Y axes with separations X and Y, respectively (note the non-standard axes orientation).

Before discussing existing viewshed algorithms and presenting our new algorithm we define some basic terms:

CELL: A rectangular or square area, $Z_{i,j}$, of size X by Y centered on $z_{i,j}$ (except at the edges or corner points: see Figure 4).

SOURCE POINT and SOURCE CELL: The point for which the viewshed is to be calculated is the source point, $s_{i,j}$. The cell corresponding to the source point is the source cell, $S_{i,j}$ (Figure 2).

DESTINATION POINT and DESTINATION CELL: The point that is to be tested to see if it is visible from the source point is the destination point, $d_{k,l}$ $(k=1,2,...,M;$ $l=1,2,...,N;$ $k \neq i$ and $l \neq j)$. The cell which a destination point represents is defined as the destination cell, $D_{k,l}$ (Figure 2).

SIGHTLINE: The line from $s_{i,j}$ to $d_{k,l}$ is the sightline, $L_{k,l}$ (Figure 2).

INTERMEDIATE POINT and INTERMEDIATE CELL: The points lying along the sightline between the source and destination points are defined as intermediate points. The cells corresponding to these points are intermediate cells.

Many algorithms have been developed for automatic viewshed computation using DEMs (Goodchild *et al.*, 1989; Floriani *et al.*, 1986; Travis *et al.*, 1975; Fisher, 1991; Sorensen *et al.*, 1993). These algorithms rely on sightlines in computing viewsheds. This means that, in order to determine whether $d_{k,l}$ is visible to $s_{i,j}$, we have to (1) set up $L_{k,l}$ from $s_{i,j}$ to $d_{k,l}$, (2) find the intermediate points between $s_{i,j}$ and $d_{k,l}$, (3) determine the positions of the intersection points of $L_{k,l}$ with the DEM grid, (4) interpolate the elevations of these intersecting points, and (5) check whether there are any intersecting points that block $L_{k,l}$. If $L_{k,l}$ is blocked by an intersecting point, $d_{k,l}$ is invisible from $s_{i,j}$ (Figure 2b). Otherwise, $d_{k,l}$ is visible from $s_{i,j}$ (Figure 2a).

These procedures are computationally intensive and hence time-consuming. Another problem is that they can overestimate the viewshed because they ignore the shape of the surface between two points of the DEM, which might also block the sightline (Figure 3). In Figure 3a, if we only consider the intermediate points, $d_{k,l}$ appears to be visible from $s_{i,j}$ because there is no point between them to block the sightline. However, they are not visible from each other because $d_{k,l}$ is not facing $s_{i,j}$. In Figure 3b, $s_{i,j}$ and $d_{k,l}$ are visible to

Department of Geography, The University of Reading, Whiteknights, P.O. Box 227, Reading, RG6 2AB, United Kingdom.

G.J. Robinson is in the NERC Environmental Systems Science Centre, The University of Reading, Whiteknights, P.O. Box 227, Reading RG6 2AB, United Kingdom.

Photogrammetric Engineering & Remote Sensing,
Vol. 62, No. 10, October 1996, pp. 1157–1164.

0099-1112/96/6210-1157$3.00/0

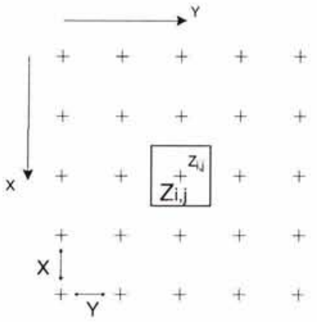

Figure 1. DEM data structure: $z_{i,j}$ is a data point of the DEM, of spacing X, Y. The rectangular area of size X by Y centered on $z_{i,j}$ represents a cell $Z_{i,j}$.

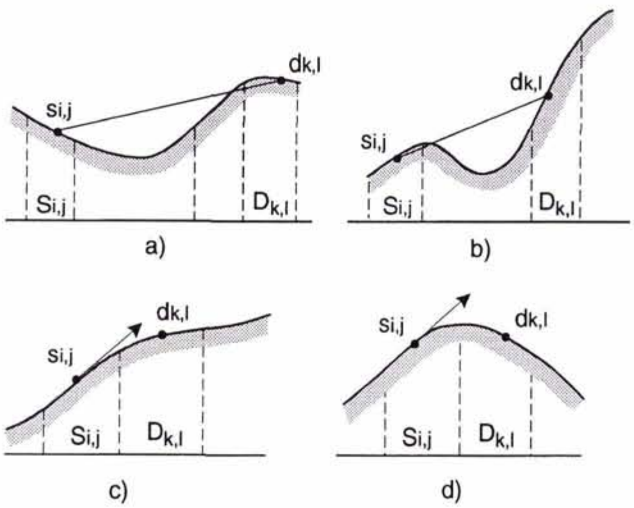

Figure 3. (a) $d_{k,l}$ is invisible from $s_{i,j}$ because the sightline is blocked by the surface between $d_{k,l}$ and its neighboring point. (b) $d_{k,l}$ is invisible from $d_{i,j}$ because the the sightline is blocked by the surface between $s_{i,j}$ and its neighboring point. (c) Although $D_{k,l}$ and $S_{i,j}$ are adjacent, they cannot "see" each other due to their difference in slope. (d) Although $D_{k,l}$ and $S_{i,j}$ are adjacent, they are invisible to each other due to their difference in aspect.

each other according to a sightline analysis, but this is not the case because $s_{k,l}$ is facing away from $d_{k,l}$. In Figure 3c, $s_{i,j}$ and $d_{k,l}$ are two neighboring points. Although there is no data point between them to block the sightline, they cannot "see" each other because of the convex nature of the surface. Similarly, in Figure 3d, $s_{i,j}$ and $d_{k,l}$ are also invisible from each other because of their differences in aspect.

The New Algorithm

To reduce the effect of the above problems, we have developed a new algorithm, consisting of five sequential steps:

(1) Detection of invisible destination points using the local surface at the source cell,
(2) Detection of invisible destination points using the local surface at the destination cell,
(3) Detection of invisible destination points using the aspect and slope of the source cell,
(4) Detection of invisible destination points using sightlines, and
(5) Detection of possibly visible destination points.

The basic approach is thus a "divide and conquer" one, in which simple tests are applied to determine the visibility or otherwise of DEM points, before having to resort to sightline analysis. These steps are now described in detail.

(1) Detection of Invisible Destination Points Using the Local Surface at the Source Cell

This step identifies invisible destination points using the planes formed by the source point and its neighbors. If $s_{i,j}$ is not at the edge or corner of the DEM, then it has eight neighboring points (Figure 4). The corresponding cell of such an an "inside point" is an "inside cell." Eight planes can be formed by the source point and adjacent pairs of its immediate neighboring points (Figure 5a): i.e.,

$$P_n\text{: } A_nx + B_ny + C_nz + D_n = 0,\ n = 1, ..., 8.$$

If $s_{i,j}$ is at a corner of the DEM, it is defined as a "corner point" and the corresponding cell is defined as a "corner cell" (Figure 4). Two planes can be formed by $s_{i,j}$ and its three neighboring points (Figure 5c): i.e.,

$$P_n\text{: } A_nx + B_ny + C_nz + D_n = 0,\ n = 1, 2.$$

If $s_{i,j}$ is at the edge of the DEM, but not at a corner, we define it as a "margin point," and the corresponding cell is defined as a "margin cell" (Figure 4). Four planes can be formed by $s_{i,j}$ and its five neighboring points (Figure 5b): i.e.,

$$P_n\text{: } A_nx + B_ny + C_nz + D_n = 0,\ n = 1,..., 4.$$

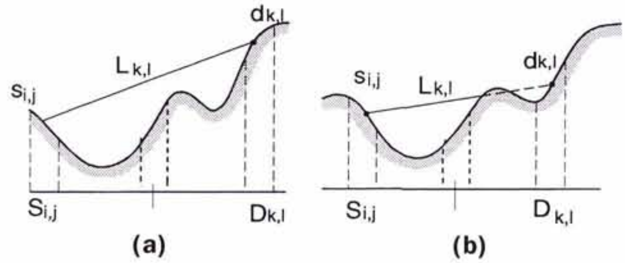

Figure 2. $S_{i,j}$ is the source cell. $D_{k,l}$ is a destination cell. $L_{k,l}$ is the sightline from $S_{i,j}$ to $D_{k,l}$. (a) Because $L_{k,l}$ is not blocked, $D_{i,j}$ is visible from $S_{k,l}$. (b) Because $L_{k,l}$ is blocked, $D_{i,j}$ is invisible from $S_{k,l}$.

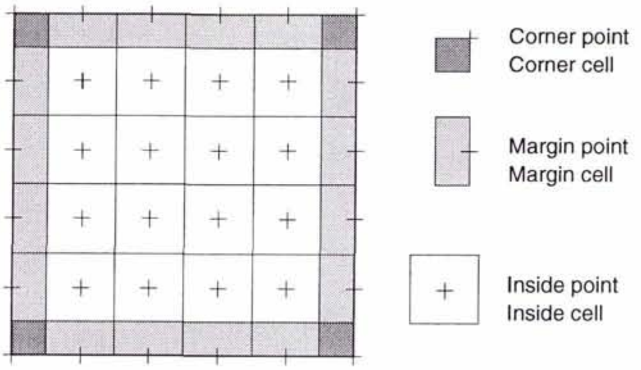

Figure 4. The possible positions of a cell in the DEM.

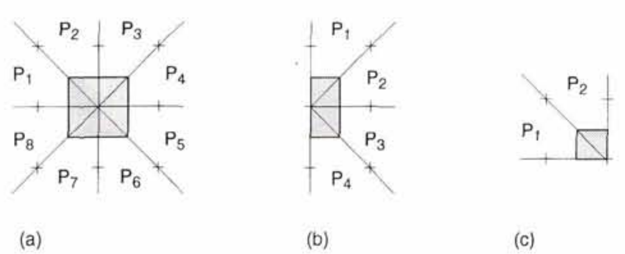

Figure 5. (a) Eight planes are defined as the reference surfaces of an inside cell. (b) Four planes are defined as the reference surfaces of a margin cell. (c) Two planes are defined as the reference surfaces of a corner cell.

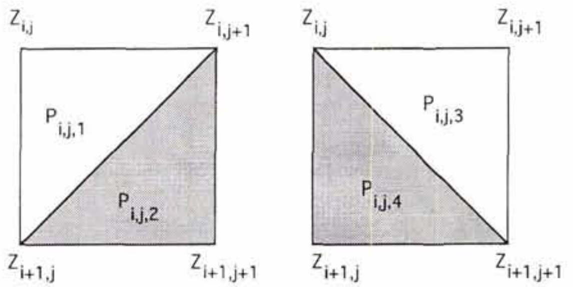

Figure 6. Four planes are generated from a grid cell of the DEM.

In calculating the viewshed for only one point in the DEM area, we use $s_{i,j}$ and its neighboring points to compute the parameters of the planes. In contrast, if we wish to find the viewshed for every point of the DEM, two planes formed by $s_{i,j}$ and $s_{i+1,j+1}$ are used twice: the first time when $s_{i,j}$ is the source point, the second when $s_{i+1,j+1}$ is the source point. To avoid duplicate calculations, we therefore calculate the parameters of the four basic planes for each cell in the DEM once only (Figure 6).

When finding the viewshed for the point $s_{i,j}$, we select the planes on the basis of it's position. For example, if $s_{i,j}$ is an inside point (Figure 5a), its eight planes are $P_1 = P_{i-1,j-1,4,} P_2 = P_{i-1,j-1,3}$, $P_3 = P_{i-1,j,1}$, $P_4 = P_{i-1,j,2}$, $P_5 = P_{i,j,3}$, $P_6 = P_{i,j,4}$, $P_7 = P_{i,j-1,2}$, and $P_8 = P_{i,j-1,1}$.

The lines formed by the intersection of each pair of adjacent planes lie along the semi-cardinal directions and divide the DEM into sectors (Figure 5, Figure 10). If $d_{k,l}$ lies in the nth sector, we use the nth plane in the analysis. If $d_{k,l}$ is below this plane, i.e., $z(d_{k,l}) \leq P_n(x_{k,l},y_{k,l})$, then it must be invisible from $s_{i,j}$.

Where the surface of $S_{i,j}$ is reasonably regular, and can be represented by a mathematical surface to an accuracy consistent with the that of the DEM, we can use this instead to identify invisible destination points. The first case is where $s_{i,j}$ and its neighboring points lie on a plane surface: i.e.,

$$P_{i,j}: A_{i,j}x + B_{i,j}y + C_{i,j}z + D_{i,j} = 0$$

where the values of the parameters $A_{i,j}$, $B_{i,j}$, and $C_{i,j}$ can be derived from the neighboring points of $s_{i,j}$ using the algorithm given by Dozier *et al.* (1990). The condition under which $d_{k,l}$ is invisible from $s_{i,j}$ is that it is below this plane (Figure 7a): i.e.,

$$z(d_{k,l}) \leq P(x_{k,l},y_{k,l}) = -(A_{i,j}\, x_{k,l} + B_{i,j}\, y_{k,l} + D_{k,l})/C_{i,j}.$$

The next case is where $s_{i,j}$ is a peak or pit and its local surface can be represented by a circular conic surface (Figures 7b and 7c): i.e.,

$$P_{i,j}: x^2/A_{i,j}^2 + y^2/B_{i,j}^2 - z^2/C_{i,j}^2 = 0$$

where the values of $A_{i,j}$, $B_{i,j}$, and $C_{i,j}$ are determined using $s_{i,j}$ and its neighboring points. Clearly, for a peak: $z \leq 0$, and for a pit: $z > 0$.

The condition under which $d_{k,l}$ is invisible from $s_{i,j}$ (Figures 7b and 7c) is then

$$z(d_{k,l}) \leq P(x_{k,l},y_{k,l}) = (x_{k,l}/A_{i,j} + y_{k,l}/B_{i,j})/C_{i,j}.$$

The neighboring points of $s_{i,j}$ are assumed to be visible from $s_{i,j}$ in this step, even though they (by definition) lie on the common plane with the source point.

(2) Detection of Invisible Destination Points Using the Local Surface at the Destination Cell

In the first step, some destination points have been determined as being invisible from $s_{i,j}$. Although these lie above the plane or the circular conic surfaces used in this step, they may still not be actually visible from $s_{i,j}$. This step checks these points.

As in the first step, if we assume that $d_{k,l}$ is the source point and $s_{i,j}$ is a destination point, we can use the planes or the circular conic surface formed by $d_{k,l}$ and its neighboring points to determine whether $s_{i,j}$ is invisible from $d_{k,l}$. If $s_{i,j}$ is below or on the surface formed by $d_{k,l}$ and its neighboring points, then $s_{i,j}$ is invisible from $d_{k,l}$, and *vice versa* (for simplification, we use only one plane formed by $d_{k,l}$ and its two neighboring points along $L_{k,l}$). If $z(s_{i,j}) \leq P_n(x'_{i,j},y'_{i,j})$ (where $x'_{i,j}$ and $y'_{i,j}$ are the coordinates of $s_{i,j}$ when the origin is moved to $d_{k,l}$. $x'_{i,j} = -x_{k,l}$. $y'_{i,j} = -y_{k,l}$), then $s_{i,j}$, is invisible from $d_{k,l}$, and *vice versa*. (Note: The neighboring points of $s_{i,j}$ are assumed to be visible from $s_{i,j}$ in this step).

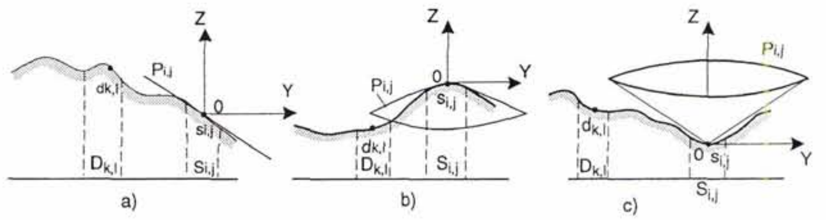

Figure 7. (a) If $d_{k,l}$ is not above $P_{i,j,}$ it is invisible from the source point. (b) and (c) If $d_{k,l}$ is not above the circular conic surface $P_{i,j,}$ it is invisible from the source cell.

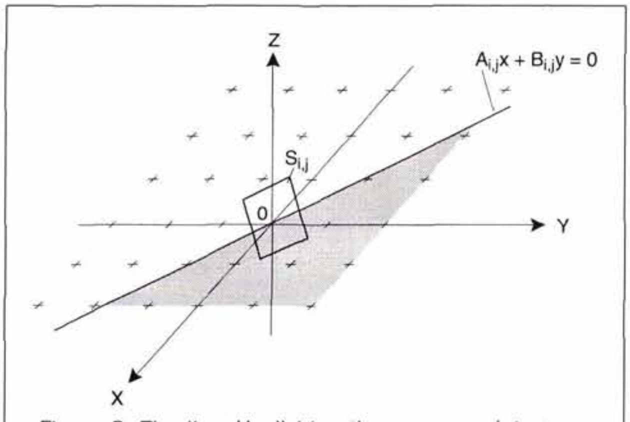

Figure 8. The line $H_{i,j}$ divides the DEM area into two parts: the shaded area is defined as the FRONT area and the other area is defined as the BACK area.

(3) Detection of Invisible Destination Points Using the Aspect and Slope of the Source Cell

This step finds invisible destination points not picked up in the previous steps. These points lie above the planes or the circular conic surfaces used in these steps, but which are still not necessarily visible from $s_{i,j}$ because of intervening points. Instead of using sightlines, we use the following method:

The intersection of the plane $H_{i,j}$ tangent to $s_{i,j}$ with the XOY plane is given by

$$H_{i,j}\text{: } A_{i,j}x + B_{i,j}y = 0.$$

$H_{i,j}$ divides the plane into two parts: the first lies above the XOY plane, the second below (Figure 8). The corresponding points of the DEM in the first part are behind $S_{i,j}$: we define this as the "BACK AREA." The points in the DEM in the second part lie in front of $S_{i,j}$: we define this as the "FRONT AREA" (shaded area in Figure 8).

If $s_{i,j}$ is a peak, all DEM points belong to the front area. If $s_{i,j}$ is neither a peak, nor a pit, nor a point lying on a plane, we can use the above algorithm to determine an approximate plane and use it to represent the surface of $s_{i,j}$. If $s_{i,j}$ is a pit, all destination points which are lower than $s_{i,j}$ have already been determined as invisible from $s_{i,j}$. (We use the algorithm developed by Peucker and Douglas (1975) to identify peaks and pits.)

Next, we divide the front area into three parts (Figure 9). Part I is the area in which every point $d_{k,l}$ and its intermediate points are lower than $s_{i,j}$. In this area, if a destination point $d_{k,l}$ is invisible from $s_{i,j}$, the intermediate point that blocks $d_{k,l}$ from $s_{i,j}$ is lower than $s_{i,j}$ and higher than $d_{k,l}$. Part II is the area where $d_{k,l}$ is lower than $s_{i,j}$ but this area does not belong to Part I. This means that there is at least one intermediate point between $d_{k,l}$ to $s_{i,j}$ that is higher than $s_{i,j}$. Part III consists of points higher than $s_{k,l}$.

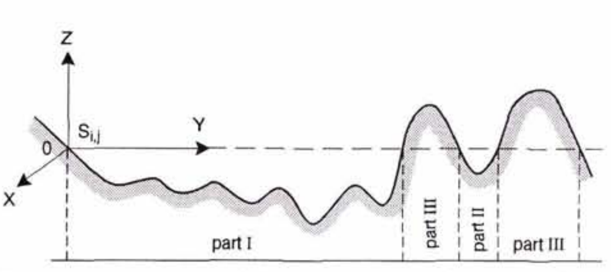

Figure 9. Subdivision of the FRONT area (refer to text for explanation).

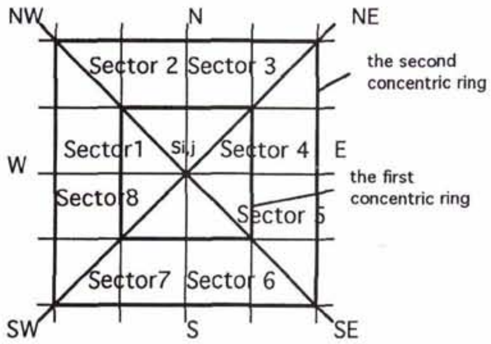

Figure 10. The DEM is divided into sectors and concentric rings centered on the source point.

The first task is to determine Part I, following which those points that do not belong to Part I and that are lower than $s_{i,j}$ must belong to Part II and are therefore invisible from $s_{i,j}$. To do this, we divide the DEM into concentric square rings (Figure 10), find points falling within Part I from the first ring of the neighboring points of $s_{i,j}$, and then proceed outwards. An auxiliary (Boolean) matrix $\mathbf{M} = \{m_{i,j}\}$ ($i = 1, 2, ..., M$; $j = 1, 2, ..., N$) is used to record the results. If a neighboring point is higher than $s_{i,j}$, then the corresponding value in **M** is set to 1 (TRUE); otherwise, it is set to 0 (FALSE). From the second ring outwards we check only the neighboring points of $d_{k,l}$. If $d_{k,l}$ is along a semi-cardinal direction from $s_{i,j}$, we need only check one neighboring point of $d_{k,l}$ that is between $s_{i,j}$ and $d_{k,l}$ (Figure 10). If the value of this neighboring point in **M** is 0 and $d_{k,l}$ is lower than $s_{i,j}$, then $m_{k,l}$ is set to 0; Otherwise, it is set to 1. If $d_{k,l}$ is not along a semi-cardinal direction, we have to check both neighboring points of $d_{k,l}$. If the values of both of these two points in **M** are 0 and $d_{k,l}$ is lower than $s_{i,j}$, then $m_{k,l}$ is set to 0; otherwise, it is set to 1. Which neighboring points should be checked depends on the position of $d_{k,l}$. As shown in Figure 10, if $d_{k,l}$ is in Sector 1, $z_{k+1,j+1}$ and $z_{k,j+1}$ should both be checked. However, if $d_{k,l}$ is in Sector 4, $z_{k+1,j-1}$ and $z_{k,j-1}$ should both be checked, and so on. Part I includes the points whose value in **M** is 0. Points that are lower than $s_{i,j}$ and that have **M** values of 1 belong to Part II (and hence are invisible from $s_{i,j}$). If $s_{i,j}$ is a pit, all destination points that are lower than $s_{i,j}$ are invisible from $s_{i,j}$ and have already been found in Step 1.

(4) Detection of Invisible Destination Points Using Sightlines

By Step 3, some destination points have been determined as invisible from $s_{i,j}$. However, the points in the back area, and Parts I and III of the front area that have not been flagged as invisible from $s_{i,j}$, need to be checked further in this step. Sightlines must be used to determine whether $d_{i,j}$ is visible from $s_{i,j}$. Because there are some points which have been determined as invisible by Steps 1 to 3, these points do not require further analysis: even if they are intermediate points, they cannot block the sightline from the source point to a destination point and therefore do not need to be considered in this step. As with Step 3, the analysis begins with the points which are near the source point and proceeds out-

Table 1. Viewshed Analysis at 10 Locations in the Snowdonia Area. Times Are CPU Seconds

Source point position (row/column)	Source point type	No. of invisible points found by Steps 1-3	Time used by new algorithm, excluding Step 5	Time used by "traditional" sightline-based algorithm	No. of visible points
300/300	x	141634	2.9	32.1	2370
200/200	x	115264	7.8	24.5	6878
50/50	x	111068	10.8	10.8	7453
398/1	x	159188	0.1	59.9	806
150/390	x	156564	0.3	42.9	23
99/360	t	157407	0.1	41.6	2314
105/363	p	121588	12.5	40.5	3069
9/383	s	126905	12.0	46.8	13003
16/8	s	82282	21.1	31.4	18612
295/358	s	150107	1.4	31.4	3036

Key: Source point type: p = peak; t = pit; s = plane surface; x = other.

Table 2. Viewshed Analysis at 14 Locations in FIFE Area. Times Are in CPU Seconds.

Source point position (row/column)	Source point type	No. of invisible points found by stages 1-3	Time used by new algorithm, excluding step 5	Time used by "traditional" sight-line based algorithm	No. of visible points
10/10	x	151323	41.3	88.2	3208
50/50	x	144842	43.1	85.3	3315
100/100	x	200829	20.2	86.3	1125
200/200	x	218735	8.0	58.1	1333
250/250	x	205017	13.5	59.1	72
400/400	x	261756	0.1	84.9	55
467/84	p	141507	44.1	91.0	916
502/420	t	262131	0.1	106.3	9
51/64	s	257660	2.5	106.1	1229
451/282	s	197843	16.4	72.2	1056
489/273	p	123609	32.8	54.7	5947
500/500	p	257517	1.2	115.6	224
231/30	p	260753	0.2	84.2	261
415/335	p	262133	0.1	80.9	9

Key: Source point type: s = plane surface; p = peak; t = pit; x = other.

wards. For every point to be checked, we find its nearest intermediate visible point. If it blocks the sightline, then this destination point is invisible from $s_{i,j}$.

When every destination point has been checked, we integrate the surface area of the visible points, yielding the viewshed area of $s_{i,j}$.

(5) Detection of Possibly Visible Destination Points

The viewshed results obtained in Steps 1 to 4 are the same as those that would be obtained using standard sightline-only based algorithms. The viewshed may therefore still be overestimated in the cases shown in Figure 3. In this step, we deal with these special cases.

We start with the neighboring points of $s_{i,j}$. In Steps 1 and 2, these points are assumed to be visible from $s_{i,j}$. But, applying the visibility criteria from Steps 1 and 2 strictly, they will be flagged as being invisible from $s_{i,j}$ because they lie on the planes which they themselves partly define. However, as Figures 3c and 3d show, whether they are visible or not is actually determined by the curvature of the surface between them, i.e., the neighbors will be visible from the bottom of a pit but not from the top of a peak. To resolve this problem, we use two pairs of planes to determine the status of a neighboring point. The first pair A and B are formed from $s_{i,j}$ and the points on the opposite side of the destination point (Figure 11). The second pair C and D are formed similarly, exchanging the source and destination points. If $s_{i,j}$ is below or on planes C or D and $d_{k,l}$ is above or on planes A or B, then this destination point is invisible from $s_{i,j}$. If $s_{i,j}$ is below or on planes C or D but the destination point is above planes A or B, or $s_{i,j}$ is above planes C or D but $d_{k,l}$ is below or on planes A or B, then $d_{k,l}$ is defined as a "possibly visible" point.

A similar method is used for other destination points that are not adjacent to $s_{i,j}$. We use the planes C or D to determine whether $d_{k,l}$ is a "possibly visible" point. If $s_{i,j}$ is below or on planes C or D, then $d_{k,l}$ is a "possibly visible" point. We only check the destination points that are flagged as visible in Steps 1 to 4, for which at least one of their neighboring points used to form the planes C or D is invisible from $s_{i,j}$.

By applying Step 5 after both Steps 1 and 2, instead of only after Step 4, it is possible to make a net saving of computation time. The value of this clearly depends on the extra time taken to run Step 5 twice being less than the time saved in eliminating additional points prior to running Steps 1 to 4.

The Experiment

We used two grid-based DEMs in a series of experiments to test the new algorithm. The first is a 400 by 400 DEM, with a grid interval of 25 m by 25 m, of Snowdonia in the UK (Figures 12 and 13). This is a mountainous area, with elevation ranging from 3 m to 734 m, and was chosen to provide significant differences in elevation, slope, and aspect and a large number of peaks and pits. The second is a 512 by 512 DEM, with a grid interval of 30 m by 30 m, of the FIFE (First ISLSCP Field Experiment) test area, in Swede Creek, Kansas. This area was chosen because it has large relatively flat

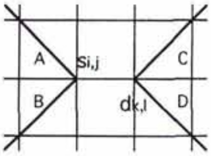

Figure 11. The planes A, B, C, and D are used to detect the "possibly visible" points in the new algorithm.

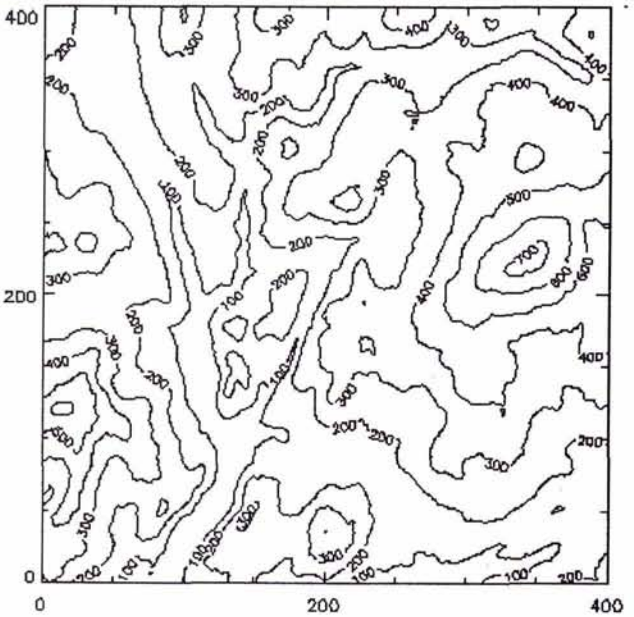

Figure 12. The Snowdonia DEM area in contour form.

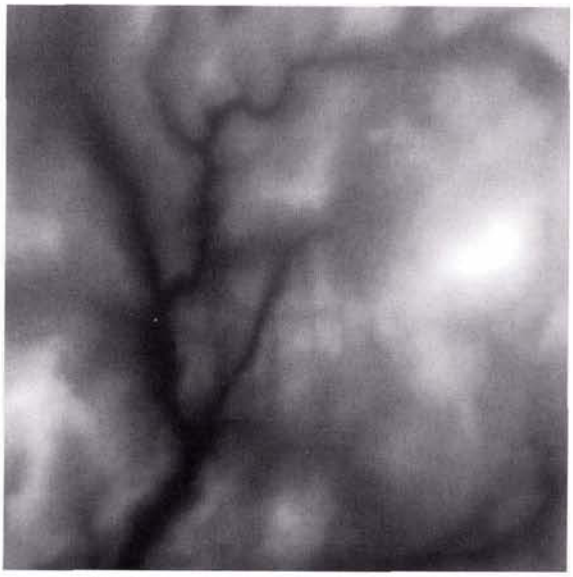

Figure 13. Grey-scale representation of the Snowdonia DEM.

regions, with elevation ranging from 309 m to 464 m (Figures 14 and 15).

Using programs written in "C" running on a Sun SparcStation 10/30 we computed viewsheds at a number of positions in both test areas using Steps 1 to 4 of the new algorithm and also with a standard sightline-based algorithm. The results, presented in Tables 1 and 2, show that overall the new algorithm appear to reduce significantly the computation times (measured in CPU seconds) compared with the standard sightline-based algorithm. (The times shown in Tables 1 to 3 do not include the time used for the calculation of the terrain parameters and planes of the cells; searching for peaks, pits, and plane surface cells; or for input/output). In addition, the greater the number of destination points that can be determined as invisible using Steps 1 to 3, the greater apparently are the time savings. For example, from the source point at $i = 99$, $j = 360$ in the Snowdonia test area, 157,407 points are invisible, with only 0.1 CPU seconds needed for the viewshed calculation using the new algorithm, compared with 41.6 seconds for the standard sightline algorithm.

How many points can be determined as invisible using Steps 1 to 3 depends on the position and orientation of the source point. For example, the point $i = 295$, $j = 358$ in the Snowdonia test area has 150,107 points which were determined as invisible using Steps 1 to 3. The surface of this cell is a plane with an aspect angle of 45° and a slope angle of 6.5°. As a result, most destination points are behind the plane and are thus flagged as invisible in Step 1. The viewshed calculation for this point used only 1.4 seconds, compared with 31.4 seconds for the standard sightline algorithm. In contrast, the point $i = 16$ and $j = 8$ has the same aspect and slope, but only 82,282 points were determined as invisible because most destination points are in front of its surface plane. The viewshed calculation for this point took 22.1 seconds compared with 31.4 seconds for the standard sightline algorithms.

For the Snowdonia test area, we also compared the results of the new algorithm, with and without Step 5. The results are presented in Table 3, which show that some 5 percent of the total points visible at any source point may not be actually visible. Possible savings in computation times by applying Step 5 after both Steps 1 and 2 are questionable, with no clear correlation between the number of "possibly visible" points and visible points.

Examples of the the viewsheds computed using the new algorithm are shown for the point $i = 210$, $j = 210$ of the Snowdonia DEM (Figure 16), and the point $i = 489$, $j = 273$ of the FIFE DEM (Figure 17).

TABLE 3. VIEWSHED ANALYSIS AT FIVE LOCATIONS IN SNOWDONIA USING THE NEW ALGORITHM, WITH AND WITHOUT STEP 5. TIMES ARE IN CPU SECONDS

Source point position (row/column)	Time used without step 5	Time used including step 5	No. of visible points	No. of 'possibly visible' points
16/8	21.1	19.0	17569	1043
10/10	15.2	19.9	9281	432
200/200	7.7	6.9	6706	172
9/383	12.0	10.3	12358	645
99/360	0.1	1.8	2166	148

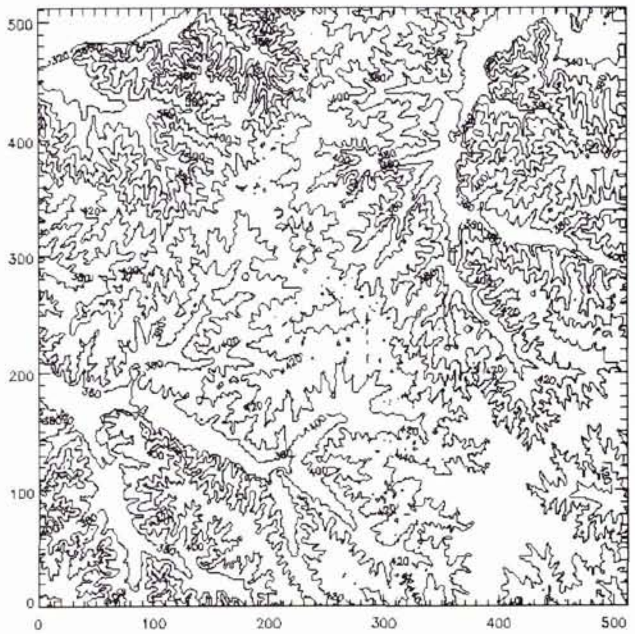

Figure 14. The FIFE DEM area in contour form.

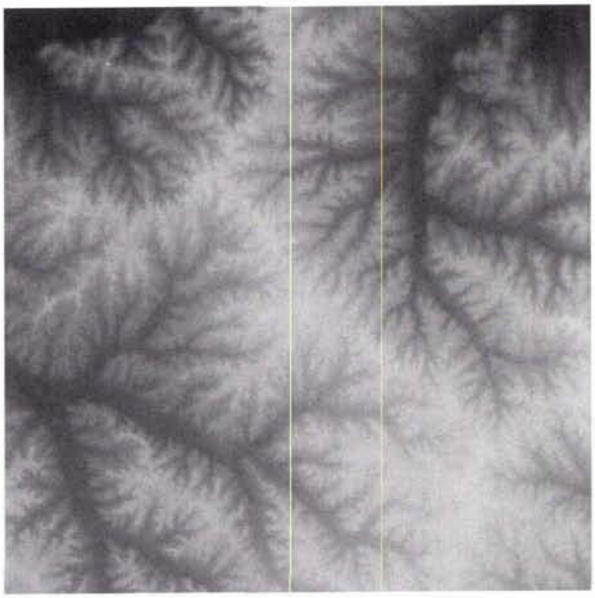

Figure 15. Grey-scale representation of the FIFE DEM area.

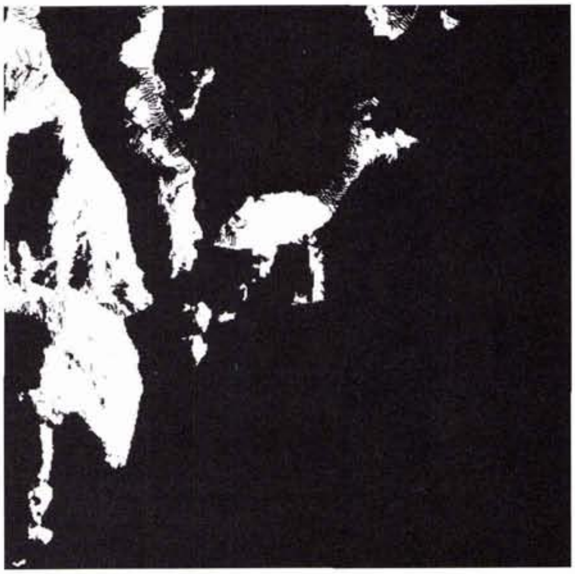

Figure 16. The viewshed of the point (i = 210, j = 210) in the Snowdonia area. The white area is visible, the black invisible.

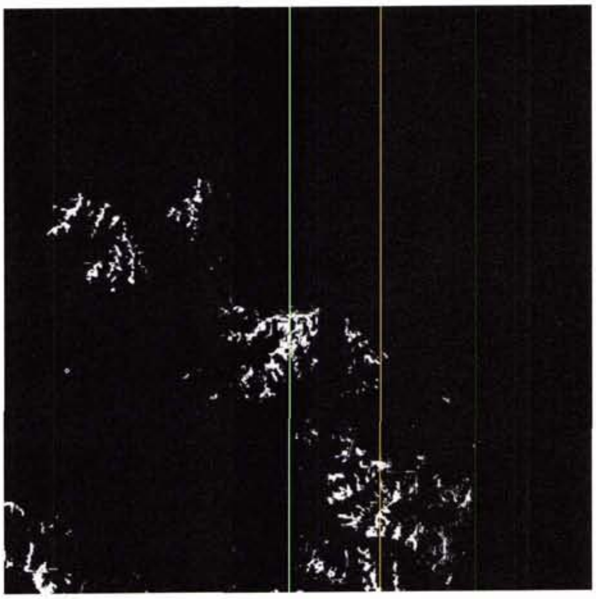

Figure 17. The viewshed of the point (i = 489, j = 273) in the FIFE DEM area. The white area is visible, the black invisible.

The time taken to find peak and pit points took only 0.8 seconds for the 400 by 400 Snowdonia DEM and 0.9 seconds for the 512 by 512 FIFE DEM. One-thousand seven-hundred sixty-three points in the Snowdonia DEM and 5,398 points in the FIFE DEM have plane surfaces. However, there are no cells whose surface can be represented by a circular conic surface in either DEM, although there are many peaks and pits in both. The calculation of the parameters for the planes or terrain parameters of each cell introduces a small overhead of 2.1 seconds for the Snowdonia DEM and 3.1 seconds for the FIFE DEM.

Conclusions

In this paper, we presented a new algorithm for automatically computing viewsheds of grid-based DEMs using five steps. The results from the first four steps are the same as those obtained using standard sightline algorithms. However, because the use of sightlines is restricted to the fourth step, which in turn exploits the results from the first three steps, significant reductions in computation time compared with solely sightline-based algorithms are possible, depending on the nature of the terrain. This makes it possible to compute viewsheds of high resolution DEMs without exhorbitant computing times. By considering the fifth step of the new algorithm, it is also possible to generate more reliable results compared with other algorithms using the same DEM by more accurately classifying points on the viewshed horizon.

Acknowledgments

The authors are grateful to the two anonymous reviewers for their helpful comments. Professor Robert J. Gurney gave continued support. The NASA/GSFC provided the DEM of the FIFE site. Dr. James McManus of the NASA/GSFC was instrumental in helping to retrieve the DEM data from the FIFE CD-ROM.

References

Dozier, J., and J. Frew, 1990. Rapid calculation of terrain parameters for radiation modelling from digital elevation data, *IEEE Trans. Geosci. Remote Sens.*, 28:960-969.

Fisher, P.F., 1991. First experiments in viewshed uncertainty: Simulating the fuzzy viewshed, *Photogrammetric Engineering & Remote Sensing*, 57:1321-1327.

Floriani, L.D., B. Falcidieno, C. Pienovi, D. Allen, and G. Nagy, 1986. A visibility-based model for terrain features, *Proceedings of 2nd International Symposium on Spatial Data Handling*, Seattle, Washington, pp. 235-250.

Goodchild, M.F., and J. Lee, 1989. The coverage problem and visibility regions on topographic surface, *Annals of Operations Research*, 18:175-186.

Jenson, E.G., and A. Rosenfeld, 1975. Digital detection of peaks, pits, ridges and ravines, *IEEE Trans. on Systems, Man, and Cybernetics*, 5:472-480.

Peucker, T.K., and D.H. Douglas, 1975. Detection of surface-specific points by local parallel processing of discrete terrain elevation data, *Computer Graphics and Image Processing*, 4:375-387.

Sharpnack, D.A., and G. Akin, 1969. An algorithm for computing slope and aspect, *Photogrammetric Engineering*, 35:247-248.

Sorensen, A.P., and L.P. David, 1993. Two algorithms for determining partial visibility and reducing data structure induced error in viewshed analysis, *Photogrammetric Engineering & Remote Sensing*, 59:1149-1160.

Travis, M.R., G.H. Elsner, D. Iverson, and C.G. Johnson, 1975. *VIEWIT: Computation of Seen Areas, Slope and Aspect for Land-Use Planning*, Technical Report, PSW-11/1975, UDSA Forest Service.

(Received 16 May 1994; revised and accepted 17 July 1995; revised 20 October 1995)

ASPRS 1996-97 Scholarships & Fellowships and Nominations for ASPRS Awards

Scholarships & Fellowships

Deadline–1 December, 1996

Robert E. Altenhofen Memorial Scholarship
To encourage and commend students displaying exceptional interest and ability in the theoretical aspects of photogrammetry.

EOSAT Award for Applications of Digital Landsat TM Data
To support remote sensing education and stimulate the development of applications of digital Landsat TM data.

William A. Fisher Memorial Scholarship
To facilitate graduate-level studies and career goals adjudged to address new and innovative uses of remote sensing data techniques relating to natural, cultural, or agricultural resources of the Earth.

Leica Inc. Photommetric Fellowship
To encourage and assist graduate and undergraduate candidates to pursue education in photogrammetry or surveying to promote the development of photogrammetric sciences.

Ta Liang Memorial Award
To facilitate research-related travel for an outstanding graduate student in remote sensing.

For eligibility, application, and more information contact:

ASPRS Scholarships/Fellowships Coordinator
5410 Grosvenor Lane, Suite 210, Bethesda, MD 20814-2160
301-493-0290 fax 301-493-0208 scholarships@asprs.org
www.asprs.org/asprs

Awards

Deadline–1 December, 1996

ASPRS Honorary Member
Recognizes individuals who rendered distinguished service to ASPRS and/or who attained distinction advancing the science and use of the mapping sciences–photogrammetry, surveying remote sensing, GIS, and related disciplines.

Outstanding Service
Given to individuals who rendered outstanding service, including professional, to the Society at the chapter, regional or national levels, helping ASPRS achieve its goals and objectives. No more than 0.05% of the Society's voting members are honored each year.

Alan Gordon Memorial
This award is given to encourage and commend individuals who have contributed to significant achievements in remote sensing and photographic interpretation.

Photogrammetric
Awarded to individuals who stimulate the development of the art of aerial photogrammetry in the U.S.

Certificate of Appreciation Meritorious Service
Recognizes special contributions to the Society that had significant impact on the overall quality of APSRS programs and activities at the national, regional or local levels.

Merit
This award is for exceptional contributions by simply serving in a specific capacity. The accomplishment requires recent and unusual time and effort of ~ 1-3 years duration. Examples of possible honorees are committee members and chairs, reviewers, regional/chapter officers, etc.

For award information/forms:
awards@asprs.org www.asprs.org/asprs