INT. J. GEOGRAPHICAL INFORMATION SYSTEMS, 1993, VOL. 7, NO. 4, 331–347

Algorithm and implementation uncertainty in viewshed analysis

PETER F. FISHER

Midlands Regional Research Laboratory and Department of Geography,
University of Leicester, Leicester LE1 7RH, U.K.

Abstract. In most documentation of geographical information systems (GIS) it is very rare to find details of the algorithms used in the software, but alternative formulations of the same process may derive different results. In this research several alternatives in the design of viewshed algorithms are explored. Three major features of viewshed algorithms are examined: how elevations in the digital elevation model are inferred, how viewpoint and target are represented, and the mathematical formulation of the comparison. It is found that the second of these produces the greatest variability in the viewable area (up to 50 per cent over the mean viewable area), while the last gives the least. The same test data are run in a number of different GIS implementations of the viewshed operation, and smaller, but still considerable, variability in the viewable area is observed. The study highlights three issues: the need for standards and/or empirical benchmark datasets for GIS functions; the desirability of publication of algorithms used in GIS operations; and the fallacy of the binary representation of a complex GIS product such as the viewshed.

1. Introduction

One issue of quality in Geographical Information Systems (GIS) which has received relatively little attention is the issue of software quality, and, of particular concern in the research reported here, the degree to which different GIS, which offer the same function, may actually report different outcomes. Essentially, different algorithms, which are designed to derive one particular phenomenon from the same data, or different implementations of a single algorithm, may give inconsistent results, and those inconsistencies may be large. The need for such testing is widely accepted outside the GIS community where such standards are easily testable. For instance, the working of much of the functionality of the word processor used to write this paper is easily tested, and is regularly so tested in reviews and comparative tests in the popular computer literature. Although Hemenway (1992), among others, makes an eloquent appeal for such benchmarking of GIS, only a limited amount has been done (e.g., Knaap 1992). In an effort to expand that literature, this paper focuses attention upon the commonly-available function known variously as the viewshed, viewable area, or visibility, and both algorithms and actual GIS implementations are compared.

The need for benchmarking is stated in the next section, and is followed by a brief review of previous work. Subsequently, the viewshed is defined, and two test data sets, which are used throughout the research reported, are introduced. The paper then goes on to examine two different issues with respect to the viewshed: algorithms for defining the viewshed are reviewed, the results of different versions coded by the author are examined, and the results of the viewshed operation from a number of different widely-available implementations are compared. The discussion presents a salutary lesson in the importance of benchmarking the precision of implementations of GIS functions, and culminates in the proposal that a probabilistic representation of the viewshed is more acceptable than the usual binary product.

2. The case for benchmarking

The operation of GIS is dependent upon a computer programmer either reading, understanding and implementing a published algorithm, or developing his/her own algorithm, to achieve a specified goal. If the programmer is lucky, he or she may even find public domain computer code to achieve the functionality required. For most complex GIS functions at least two alternative algorithms exist, and frequently many more can be found in the literature. Where numerous algorithms exist for the same function, they may make different initial assumptions and use different approximations in achieving the same goal. If two programmers sit down to achieve the same functionality, however, even if they adopt the same broad algorithm, it is unlikely that the code will look exactly the same. Parts of the implementation may be executed in alternative orders, and equations may be split up into variable numbers of lines of code in the implementation. If the two programmers are working on alternative computer-platforms, then the code may be identical, but still derive inconsistent outcomes owing to the different precisions of either compilers or hardware (Burrough 1986). Problems with a high level of algorithm complexity, therefore, are almost certain to have as many different , but correct answers as there are implementations. There is very clearly a need for study and evaluation of standards in the functionality of GIS, so that users may have confidence in the modelling they do (Hemenway 1992, Jordan and Star 1992).

3. Previous work

Some comparative studies of GIS algorithms do exist in the literature. Skidmore (1990) compared three of the many methods for mapping drainage networks from gridded DEMs as well as proposing his own. Lee (1991) studied four methods of converting from a DEM to TIN data structure, with elimination of many observations in the DEM. These are both relatively exotic operations, and only occasionally available in operational GIS. On the other hand, both Skidmore (1989) and Kvamme (1990) have studied the precision of algorithms for extracting slope and aspect from gridded DEMs, and Srinivasan and Engel (1991) have explored how that algorithm error propagates into estimates of soil erosion. Wagner (1989) has given an exhaustive review of three polygon overlay operators.

Several discussions of raster to vector conversion algorithms have occurred in the literature (e.g., Clarke 1985, Piwowar *et al.* 1990), and, unusually, this operation is the subject of Knaap's (1992) examination of implementations. Knaap reports the rather pessimistic finding that among the eight packages tested, no two produced the same rasterization of the test vector patterns, although there were some reassuring consistencies. The need for such comparisons has been largely ignored in the drive for standards in GIS which have concentrated on data transfer (e.g., DCDSTF 1988) to the almost total exclusion of any others (but see Hemenway 1992, and Jordan and Star 1992). This paper aims to further the literature in this area by presenting the results of an exploration of both algorithms and operational implementations of the viewshed operation.

4. Experimental design and test data

4.1. *The viewshed*

In all implementations examined here, the viewshed function takes a gridded Digital Elevation Model (DEM) and a single viewing point, and derives a new raster image, showing those cells which are visible from the viewing point and those which are not (coded 1 and 0, respectively). This is in conformity with the usual reporting of the viewshed. In all GIS examined here, a gridded DEM is processed. Options which exist in some of the implementations tested and others available include specification of the height of the viewer at the viewing point, the distance to which the viewer is interested, and a height of land covers in the study area which may block views (trees, walls, etc.). Less commonly it may also be possible to specify the angle of viewing (with respect to north), a height above or below which objects may not be visible, earth curvature and haze effects, although the last is usually available only for landscape visualization. In the current research the total viewable area from a specified height above a particular location over a piece of terrain unencumbered by vegetation is the desired result, this being the lowest common denominator of the different viewshed implementations examined.

Unlike some other GIS functions the viewshed is not actually verifiable in the field nor can it be logically validated, anything more than trivially, by examination of test figures, as Wagner has done for the overlay algorithm (Wagner 1989). The problem is that almost everywhere vegetation intervenes in the landscape to some height above the ground which is hard to measure with any precision. Furthermore, atmospheric refraction and earth curvature, which are both complex and rarely included in the viewshed function, cause direct lines-of-sight to be different from the actual lines viewed along. Furthermore, the database error in the DEM has a significant impact on the viewable area that can be calculated (Fisher 1991). In short, even if a precisely determined DEM were to exist with minimal vertical and horizontal errors (surveyed by GPS, for example) for a vegetation-free landscape, it is not certain that a laser determined line-of-sight between points would correspond with the viewer's line of sight. For all these reasons, the approach taken here is of comparative testing to establish the possible numerical and spatial variability in viewsheds.

4.2. *The test area*

Two test areas are used which have been employed in other research published by this author (Fisher 1991, 1992). Each area is a 100 by 100 pixel subset of the level 1 Digital Elevation Model (DEM) for the Coweeta Experimental Watershed in North Carolina (the Otto 7·5 minute quadrangle) (figure 1). Each DEM has a single viewing point associated. Maps and histograms of the elevations of the two areas are presented in figure 1, and it can be seen that one view point is chosen to be relatively low in the local landscape (in a valley position), and one high in the landscape (on a hill-top). Both locations have considerable relief in the immediate vicinity of the viewing point. The testing reported here could have been achieved with any DEM, and any viewing location on that DEM.

5. Testing algorithms

5.1. *The basic algorithm*

Two essential steps occur in the basic algorithm addressed in the research reported here. The first detects the horizontal location at which the line-of-sight (LOS) from the viewing point to the target intersects the grid of the DEM. The second compares the DEM elevation at the intersection with the elevation on the LOS. If the latter is higher, then processing can proceed to evaluate the next intersection point, otherwise the target can be declared as invisible from the viewpoint, and processing proceed to consider another target.

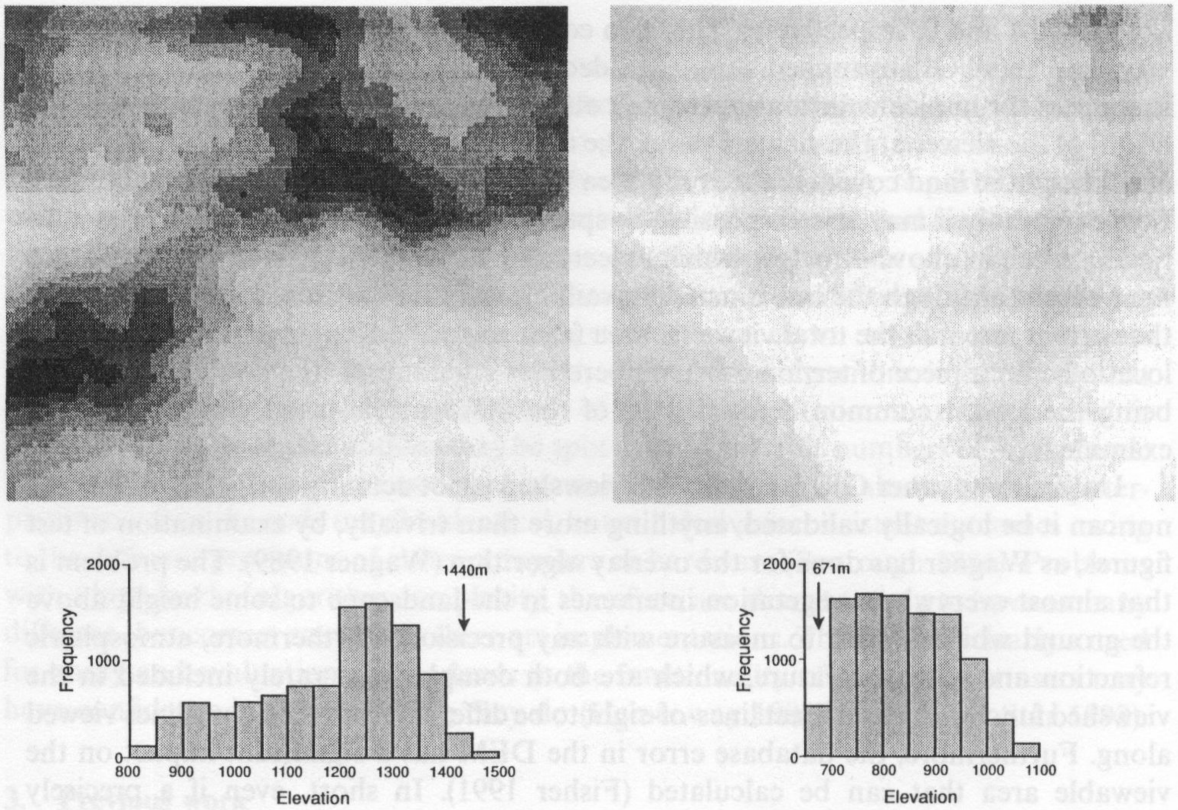


Figure 1. Shaded contour maps and histograms of the the elevations in the two DEMs used for testing. In the contour maps, higher elevations are darker, the shades do not overlap between maps, and the viewpoints are indicated as isolated black pixels. In histograms viewpoint elevations are shown by arrows.

The method for inferring elevations within the grid and determination of the number and positions of lines for comparison with the LOS is profoundly important in this process. Similarly, the way in which the grid cell structure is interpreted with respect to the viewing point and the target, and, indeed, the whole DEM has clear importance. The second step, when the elevations are calculated, is also crucial to finding the viewshed, because this is the decision step.

Some very different algorithms have been suggested (e.g., Mark 1987, Teng and Davis 1992), but they are not explored because they yield only approximations, and because the research which is reported here makes a significant point without confusing the issue by branching into these alternatives.

5.2. *Inferring elevations*

From a review of the literature, and discussions with individuals implementing the viewshed operation, it has been possible to identify four different methods for inferring elevations from the basic DEM. These are either documented, suggested or implied.

The first method is to take each elevation in the DEM to be at the centre of a grid cell, and to infer continuous, linear change to the centres of each of the four neighbours of that grid cell, the DEM allowing inference of a mesh of sloping lines. The LOS from the target location is calculated to the intersection of each of the mesh lines between neighbours (figure 2 (*a*)), and heights compared at the target. If a second viewpoint is chosen then the mesh of elevations remain the same (figure 2 (*b*)). This method is specified by Yoeli (1985).

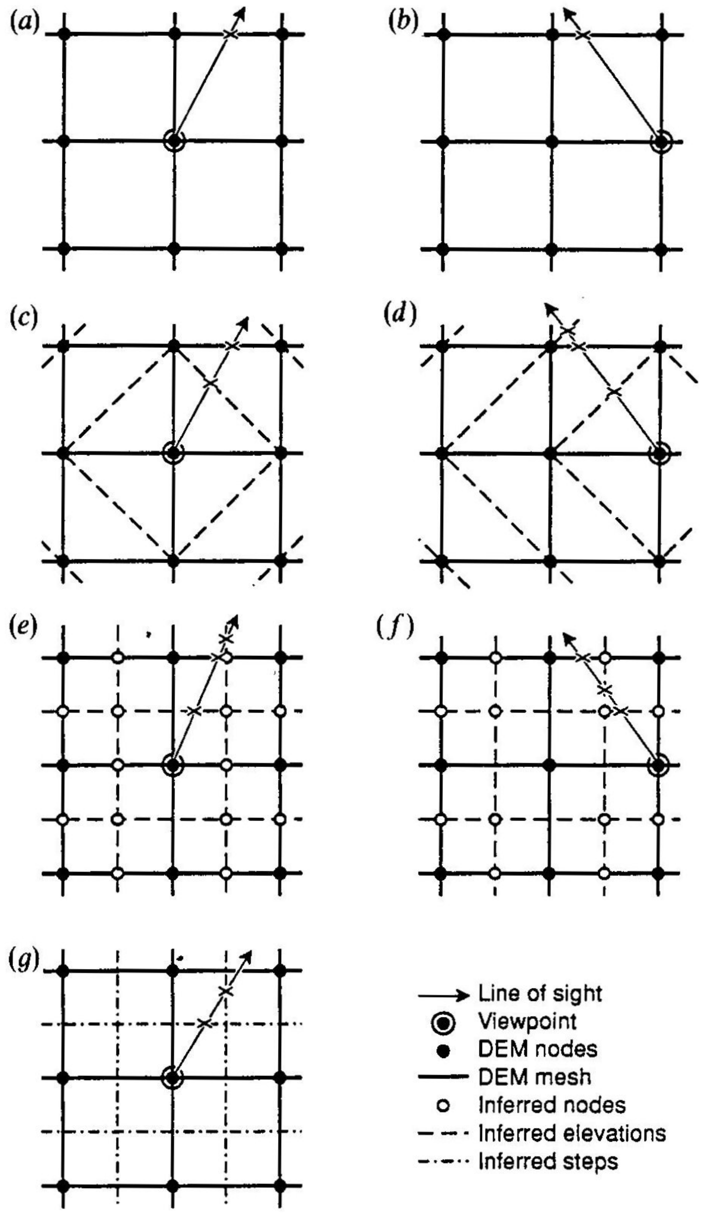


Figure 2. Alternative methods of extrapolating from the elevation values in a DEM to an inferred network of elevations, including (*a*) and (*b*) linear interpolation between grid neighbours, (*c*) and (*d*) triangulation of the grid, (*e*) and (*f*) grid constraint of the mesh, and (*g*) the stepped model.

A second method, which can be carried over from the TIN data structure, is to triangulate the DEM, inferring a connection from one cell to the diagonal neighbours as well as the orthogonal neighbours. Eight lines of continuously changing elevation from each grid cell are possible. Use of all eight would mean that, at the crossing of the two diagonals within a cell, there would be very likely to be two elevations. This would not be acceptable, and so the triangulation is executed as illustrated in figure 2 (*c*) and 2 (*d*). The characteristic of this is that in moving from one viewing point to another the elevation network to be compared changes, since the diagonals change.

Tomlin (1990) suggests the imposition of a secondary grid on the first, where new elevations are inferred equidistant between pairs of neighbours and four neighbours. On original grid lines the inferred elevations are the mean of the two neighbours, while at the centres of original mesh squares they are the mean of the four neighbours (figure 2 (*e*)). Moving the viewing point does not change the network of elevations (figure 2 (*f*)).

These first three models increasingly constrain the possible viewshed, because the number of points at which the LOS to the target will be compared with the LOS to the mesh location increases before any point can be shown to be in view. It therefore follows that in any situation the viewshed is likely to be progressively smaller from the first to the third.

Finally, an elevation model may be regarded as stepped. Each elevation is applied to the whole grid cell, with four vertical faults at the edges of the cell where elevations change to the next cell (figure 2 (*g*)). This model is not only used in viewshed calculation (Felleman and Griffin 1990, Teng and Davis 1992) but is widely used in illustrating other algorithms for manipulating DEMs (e.g., Travis *et al.* 1975, Burrough 1986).

5.3. *Results for elevation inference*

Each of the four foregoing approximations of elevation models was implemented, and run for the two test viewing locations. The results are presented in table 1. It can be seen that in the cases of both viewpoints, the area of the viewshed decreases from the simple grid to the stepped grid. The final step from the grid-constrained model to the stepped model is the most dramatic, but that from the triangular to the grid constrained is also large. Futhermore, the degree of change is not consistent for both viewpoints, and so may be dependent on the landscape position.

The pattern of cells included in the various viewsheds derived is presented in figure 3. It can be seen that generally the patterns are nested within each other. In detail, however, this observation does not hold. Some methods have identified cells as being out-of-view, when they are in the heart of the viewable areas defined by other methods, and an examination of the frequency data presented in table 1 shows that no nesting is really occurring; if it were the number of cells with the highest cell count (four in this case) should be equal to the smallest viewshed, and the total number of cells included in

Table 1. The number of cells included in the viewsheds from the test locations. Results for the four different methods for inferring elevations discussed in the text, and illustrated in figure 2, are given.

Elevation approximations	Test site 1	Test site 2
Grid	2381	2034
Triangular constraint	2312	1917
Grid constraint	1992	1442
Stepped	656	92
Frequency of cells in summed image (figure 3)		
5 (Viewing point)	1	1
4	604	72
3	1409	1357
2	324	513
1	46	96
Total	2384	2038

N.B. the first entry is the same as in tables 2 and 3.

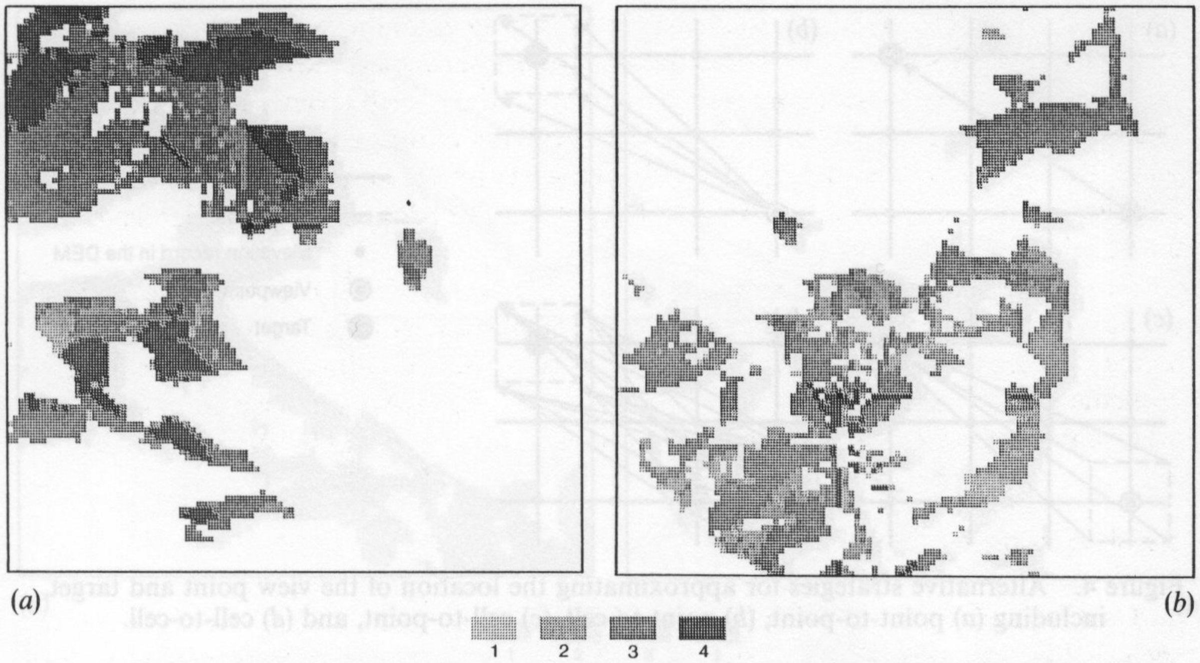


Figure 3. The pattern of grid cells included in viewsheds derived from alternative methods of inferring elevations.

all viewsheds should equal the largest of the original viewsheds. In the event, the number of most frequent cells is considerably smaller than the number of cells in the smallest viewshed reported. On the other hand, the total number of cells is very nearly the same as the largest, except for a few cells. There is therefore some mixing of pixels in- and out-of-view in the single viewsheds.

5.4. *Viewer and target locations*

Specification of how the viewer and target locations should be treated within the viewshed operation can also cause considerable variation in the viewable area. The grid models of the DEM discussed above suggest the approximation of the viewing and target locations as nodes of the DEM mesh; although some other point may be the viewing location, it would not be logical for the target to be other than the node as representative of the raster grid cell for which visibility is to be reported. This Point approximation is, however, not implicit in the raster data structure to which the DEM is tied, and in which format the viewshed is reported. Instead, a grid cell approximation might be taken and the visibility to or from the whole area of the grid cell examined. The stepped approximation of elevation inference effectively recognises all cells as grid cells. Recognition of two alternative interpretations of a grid cell yields four different possible combinations of viewing and target locations: Point-to-Point (figure 4 (*a*)), Point-to-Cell (figure 4 (*b*)), Cell-to-Point (figure 4 (*c*)), and Cell-to-Cell (figure 4 (*d*)).

Exactly how the Line-of-Sight should be found for approximations involving cells is not clear, because technically every single possible viewing point on the cell should be explored, but there is an infinite number. Here it is assumed that the LOS from the corners of the viewing cell or to the corners to the target cell will determine whether a target is visible. In the case of cell-to-cell, this leads to 4^2 separate comparisons for any one target. This is of course reduced, because once the target is declared visible from one corner of the viewing cell, then it is in-view, and processing can proceed to consider the next case; but the target cell can only be declared as out-of-view after 16 LOS calculations. In all tests the simple grid method for inferring elevation is used.

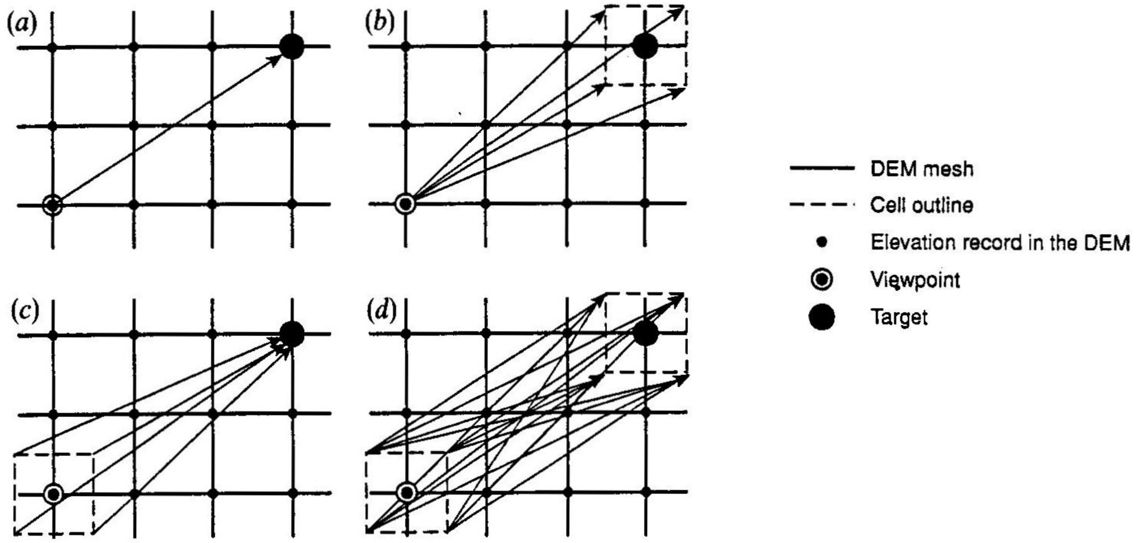


Figure 4. Alternative strategies for approximating the location of the view point and target, including (*a*) point-to-point, (*b*) point-to-cell, (*c*) cell-to-point, and (*d*) cell-to-cell.

5.5. *Results of viewpoint and target approximation*

As might be expected, the Point-to-Point yields the smallest viewshed and the Cell-to-Cell the largest (table 2). The changes can be dramatic, with an increase of nearly 50 per cent being recorded in the area viewable from viewing point 1 between the Point-to-Point and Cell-to-Cell cases. The viewsheds might be expected to be nested in this case, but again this is not precisely the case, although in both cases the total number of cells with some degree of visibility in the sum image (figure 5) is very close to the total for the cell-to-cell comparison.

Interestingly the cell-to-point and cell-to-cell analyses of the viewshed for test 1 are the only ones in all the tests reported here to identify a whole area to the bottom/centre of the image as being in view. The very straight, north-east facing edge of this area implies that it is just visible over a ridge.

Table 2. The number of cells included in the viewsheds from the test locations. Results for the four different methods for approximating viewer and target locations and illustrated in figure 5, are given.

Viewing point	Test site 1	Test site 2
Point to point	2381	2034
Cell to point	3328	2271
Point to cell	2707	2666
Cell to cell	3970	2907
Frequency of cells in summed image (figure 5)		
5 (Viewing point)	1	1
4	2350	2014
3	129	176
2	1069	545
1	457	200
Total	4006	2935

N.B. the first entry is the same as in tables 1 and 3.

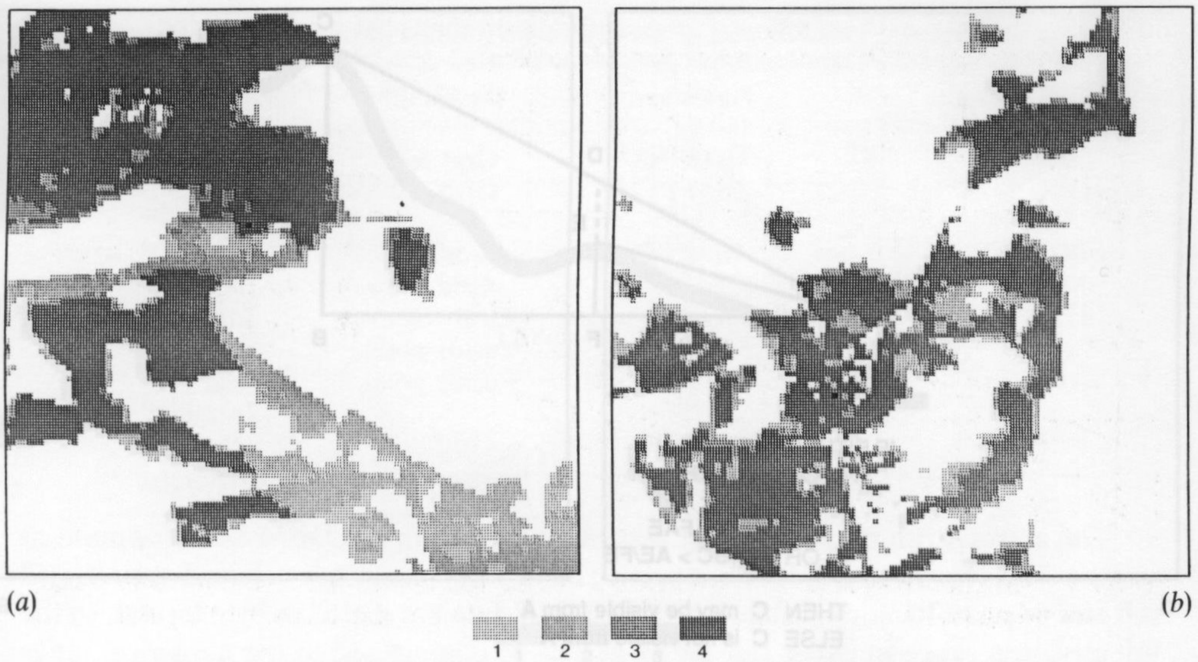


Figure 5. The pattern of grid cells included in viewsheds derived from alternative methods of view point and target approximation.

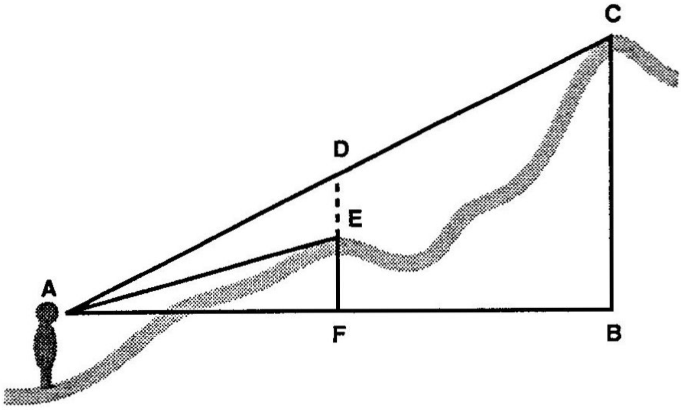
5.6. *Comparing elevations*

Once the location at which the elevation should be tested is determined, there are actually several ways in which the judgment as to the relative elevations of the two lines can be made (figure 6). Given that ultimately the vertical side of two right angle triangles are being compared, at least the five simple tests listed in figure 6 can be evaluated to yield a correct answer, although they are, of course, all interrelated and *should* give the same answer. These five possible tests are: the lengths of the vertical lines (the relative height), the hypotenuses, or either of the two non-right angles of the triangle may be tested, as may the gradients of the line-of-sight to the target and to the interim location. The angles may be evaluated in either a trigonometric form or in degrees or radians. The interesting point is that the result should be the same whichever is tested, and usually will be so long as the difference in height is large; when the difference is small rounding errors in the compiler and/or the processor may create inconsistencies. Furthermore, when the two test heights, gradients, etc., are equal, the target should be reported as out-of-view, but it would be a simple error in coding for this not to be the case; probably an error with slight, but measurable consequences.

To test the significance of the precision, three different versions of the elevation comparison were implemented. In the first, the height at the intersection of the LOS and the grid was tested (this test was used in all the inferring comparisons above), in the second the gradient of the two lines was compared, and in the last the final result and all interim calculations were rounded to 16 bit integer values without scaling (to simulate Fortran integer arithmetic which is used in some systems to speed processing). In all cases the simple grid inference, and point-to-point viewing was used

5.7. *Results of elevation comparisons*

The results are presented in table 3. Differences are noted between all three methods, but they are not as large as in the preceding experiments. A consistent pattern



IF EITHER AD > AE
OR DF > EF
OR $A\hat{C}B < A\hat{E}F$
OR $B\hat{A}C > F\hat{A}E$
OR AC/BC > AE/FE

THEN **C** may be visible from **A**
ELSE **C** is not visible from **A**

where **A** is the viewing position at the specified elevation
B is the horizontal position of the target location which is at elevation **BC** above **A**
F is the horizontal position of the current location which is at elevation **FE** above **A**

Figure 6. Five alternative methods for evaluating elevations discussed in the text when the elevation of the target is higher than the viewer. If the elevations are reserved, then the geometry is different, and the conditional statements listed are reversed.

Table 3. The number of cells included in the viewsheds from the test locations. Results for three of the different methods discussed in the text, and illustrated in figures 6, are given.

Comparisons	Test site 1	Test site 2
Height	2381	2034
Gradient	2310	1877
Integer height	2553	2052
Frequency of cells in summed images (figure 7)		
4 (Viewing point)	1	1
3	2270	1794
2	104	223
1	223	132
Total	2598	2150

N.B. the first entry is the same as in tables 2 and 3.

can be discerned, however. The integer rounding always produces the largest area, and the gradient calculation yields the smallest. No particular reason is apparent for this.

Again the pattern of cells appears to show nested viewsheds (figure 7), but examination of frequencies in the image shows considerable mixing in the viewsheds (table 3).

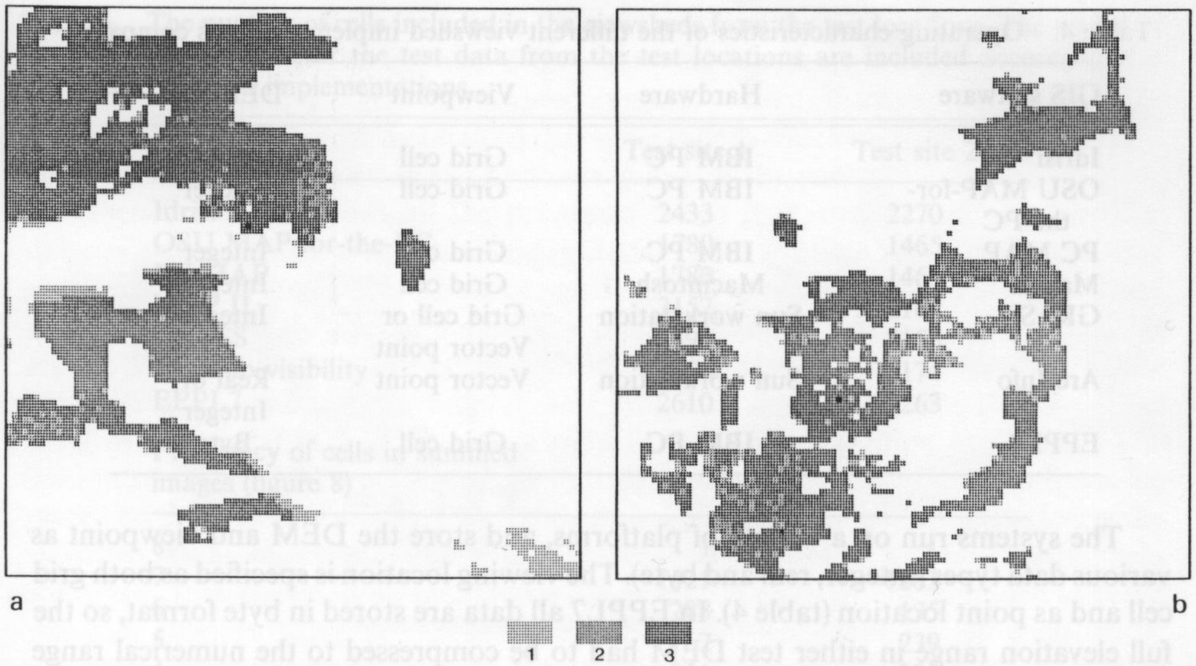


Figure 7. The pattern of grid cells included in viewsheds derived from three alternative methods of comparing elevation.

5.8. *Other possible causes of variation in the viewshed*

The results presented show considerable variety in the area of the viewshed derived. Although not attempted here, similar variations to those found in the third test might result from moving platforms owing to variable processor precision, or changing compilers or languages because of different methods of handling interim values in formulations, and because of different precisions in handling floating point numbers. In evaluating a single line of code some compilers store interim calculations to very high precisions (the highest allowed by the operating system or processor), lowering the precision to the level for the variable declared (integer, real, double, etc.) only at the last stage, while others limit the precision to the final precision throughout. Of course, the precision is also dependent upon the hardware, and the code on one platform (with, say, 16-bit precision) may yield a different result from another (with 32-bit precision).

6. Comparing implementations

6.1. *The GIS*

With the cooperation of a large number of individuals (academics, system developers, and some who are both, see list of acknowledgments) it has been possible to run the data through a number of different GIS packages all purporting to report the viewable area. Table 4 presents some of the characteristics of seven systems for which the viewshed was tested. Most of the systems are in the educational domain owing to the availability of software and the willingness of colleagues to give their time. Data were supplied to those testing the programs in the Idrisi ASCII format, with the DEMs and viewpoints being separate image files. Participants were all asked to read the data into their own GIS, to execute the viewshed, forcing the program to report for the whole area, and placing the viewpoint at 2 m above the DEM surface. The resulting viewsheds were returned to the Idrisi ASCII format and combined under that GIS. Some GIS systems were excluded from the comparison because of the apparent lack of the functionality to allow export to any useful format.

Table 4. Operating characteristics of the different viewshed implementations compared.

GIS software	Hardware	Viewpoint	DEM file
Idrisi	IBM PC	Grid cell	Real
OSU MAP-for-the-PC	IBM PC	Grid cell	Integer
PC MAP	IBM PC	Grid cell	Integer
Map II	Macintosh	Grid cell	Integer
GRASS	Sun workstation	Grid cell or Vector point	Integer
Arc/Info	Sun workstation	Vector point	Real or Integer
EPPL7	IBM PC	Grid cell	Byte

The systems run on a variety of platforms, and store the DEM and viewpoint as various data types (integer, real and byte). The viewing location is specified as both grid cell and as point location (table 4). In EPPL7 all data are stored in byte format, so the full elevation range in either test DEM had to be compressed to the numerical range 0–255. Since the viewer elevation also has to be in byte, it is not possible to make the viewer elevation exactly comparable with the elevation in other tests. The result is not strictly comparable with others, therefore, but it is included for completeness. In the Arc/Info Visibility operation, the viewing point is specified as a vector point, and this was digitized within Arcedit as close to the mid-point of the viewing grid cell used in the other tests as possible.

6.2. *Results of implementation comparisons*

The results are striking (table 5, figure 8). PC MAP and OSU-MAP-for-the-PC are the only two programs to derive exactly the same viewshed. The fact that the latter is based on the former, with enhancements to the user interface and functionality, means that they are almost certainly using not only the same code, but probably the same compiler. The fact that these yield identical viewable areas, which are smaller than any other in both instances, suggests that the grid constraint is imposed, which is also not surprising since it was suggested by Tomlin (1991 p. 35), the developer of PC MAP.

All other programs have produced different results, but with consistently larger areas than MAP. Assuming that all implementations are based on the LOS algorithm adopted here, it would appear that elevations are inferred by either the grid or triangulation method, and that all use point-to-point viewpoint-to-target approximation, because the areas of the viewsheds are closest. Nothing can be said about the elevation comparison method used, because the variation in the controlled experiments is less than in the implementations.

Only one gross error was found in the testing, and this has been reported to the developers. It causes a slight overestimate of one viewable area, and is easily identified in figure 8 (*b*) as a horizontal line running right across the image, through the viewpoint. The same error occurs in the viewshed for point 1, but it is not so obvious in figure 8 (*a*), because the erroneously identified cells are not continuous across the image.

7. The probable viewshed

The variability of the viewshed, caused by changing minor parameters in the viewshed operation, highlights the inappropriateness of the binary representation of the viewshed. It is believed that the probable viewshed yields a more appropriate

Table 5. The number of cells included in the viewsheds from the test locations. The viewshed areas calculated for the test data from the test locations are included according to 8 different GIS implementations.

GIS software	Test site 1	Test site 2
Idrisi	2433	2270
OSU MAP-for-the-PC	1780	1465
PC MAP	1780	1465
MAP II	2157	2161
GRASS	2390	2156
Arc/Info visibility	2304	2174
EPPL7	2610	2263
Frequency of cells in summed images (figure 8)		
8	1	1
7	1421	1283
6	288	135
5	162	239
4	351	335
3	118	173
2	150	226
1	431	493
Total	2922	2885

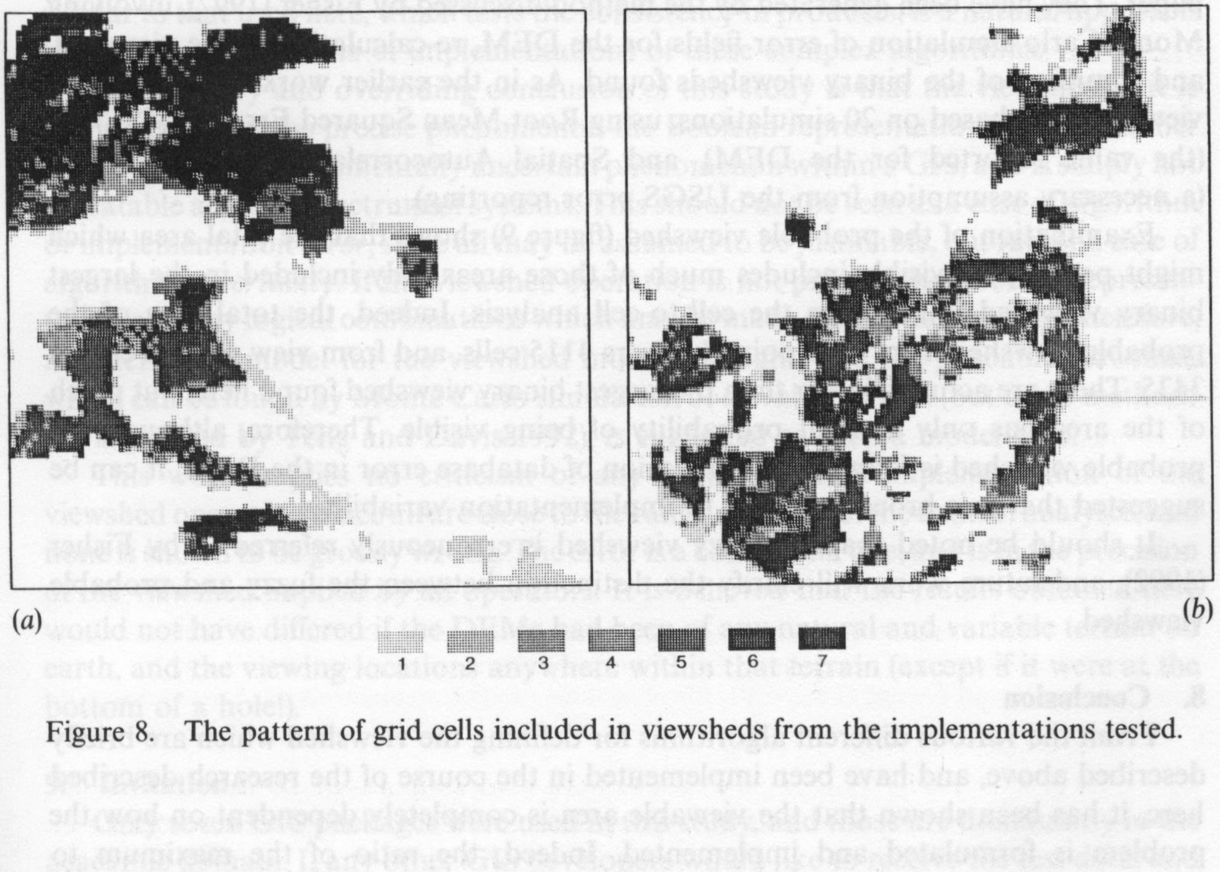


Figure 8. The pattern of grid cells included in viewsheds from the implementations tested.

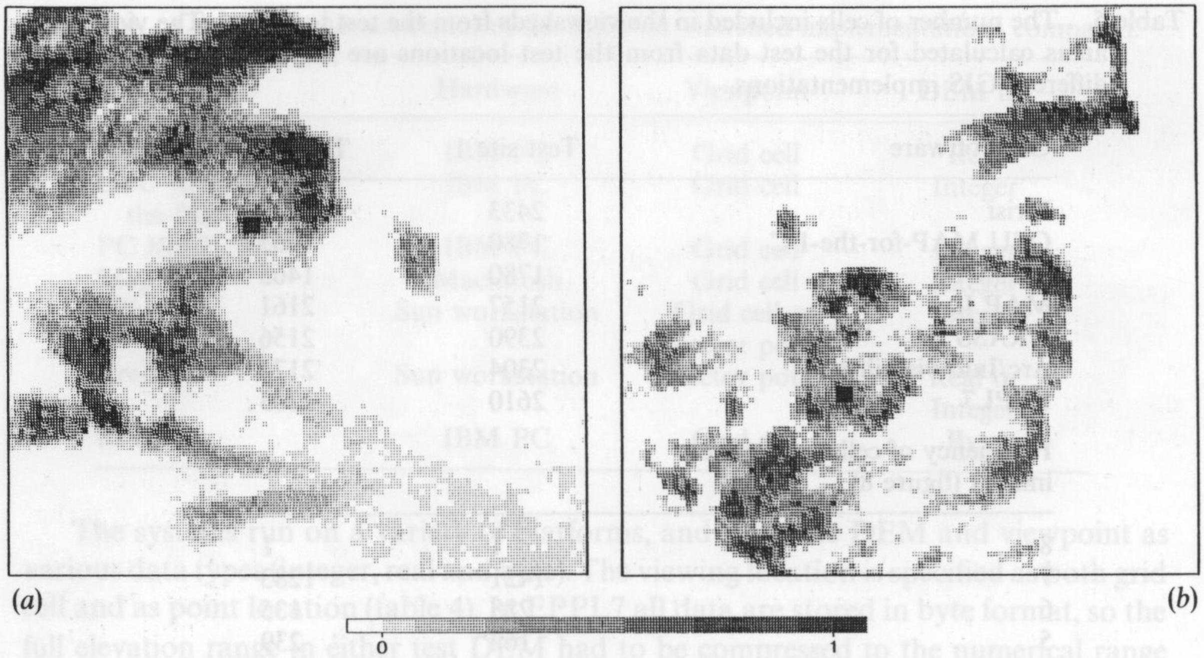


Figure 9. The pattern of probabilities of cells being visible from the viewing points, as defined by repeatedly simulating error in the DEM, and adding the binary viewsheds found together (Fisher 1992).

model. It recognises the Boolean nature of the viewshed, but for any location there is not a report that it is simply in or out of sight; rather a likelihood that it can be seen is given.

Figure 9 shows the probable viewsheds for the two test areas used throughout this paper. They have been generated by the method discussed by Fisher (1992), involving Monte Carlo simulation of error fields for the DEM, re-calculation of the viewshed, and summing of the binary viewsheds found. As in the earlier work these probable viewsheds are based on 20 simulations, using Root Mean Squared Error (RMSE) = 7 (the value reported for the DEM), and Spatial Autocorrelation (Moran's I) = 0 (a necessary assumption from the USGS error reporting).

Examination of the probable viewshed (figure 9) shows that the total area which might possibly be visible includes much of those areas only included in the largest binary viewshed found from the cell-to-cell analysis. Indeed, the total area of the probable viewshed from view point 1 covers 4115 cells, and from view point 2 covers 3435. These are actually larger than the largest binary viewshed found here, but much of the area has only a slight probability of being visible. Therefore, although the probable viewshed is found from simulation of database error in the DEM, it can be suggested that it is broadly related to implementation variability.

It should be noted that the fuzzy viewshed is erroneously referred to by Fisher (1992), and future work will clarify the distinction between the fuzzy and probable viewshed.

8. Conclusion

From the various different algorithms for defining the viewshed which are briefly described above, and have been implemented in the course of the research described here, it has been shown that the viewable area is completely dependent on how the problem is formulated and implemented. Indeed, the ratio of the maximum to minimum viewable area is over 6 and 31, for viewpoints 1 and 2 respectively, comparing cell-to-cell to stepped, or around 2 for both test locations if the second smallest viewable area, from the grid constrained method, is used. Furthermore, it has been shown that all actual implementations derive viewsheds which are of an area within the areal and spatial limits bounded by those coded in the course of research, but are nonetheless variable. Thus all the programs tested (bar one) seem to be 'correct'; differences result from alternative design decisions at the time that algorithms were specified by different groups. Those design decisions were undoubtedly defensible and acceptable both at the time and now. The point is that many decision stages exist for any implementation of a viewshed algorithm, and designs may diverge and cause inconsistent viewsheds to be found (unless programs share code as in PC MAP and OSU MAP-for-the-PC). Users have a right to know the algorithm used in any one version of the viewshed operation, so that they may assess how much faith can be placed on the results of an analysis. If the cell-to-cell algorithm is used, then clearly the viewshed is not so acceptable, but perhaps if point-to-point is employed using grid constraint, then considerable confidence can be placed on locations identified as being viewable actually being so. Insufficient testing has been done to establish which algorithm gives the best results in any particular circumstance, even if that were possible. Future research will attempt to examine the relative performance of the different algorithms under different landscape conditions.

The viewshed is only one of a number of complex functions contained in many GIS which have different possible algorithms. There is a very clear need for extensive comparative testing of GIS functions, as is reflected in several recent appeals for software standards (Hemenway 1992, Jordan and Star 1992). It is preferable to use simple geometric objects where the result is completely predictable, but in many cases of GIS functions (such as the viewshed) this is not appropriate. An experimental design similar to that used here, which tests the consistency in products, is a natural approach to analysing the results of implementations of these complex algorithms.

The primary and overriding conclusion of this study is that the viewshed as it is calculated is not the precise phenomenon the Boolean representation implies. Rather the viewshed is a fundamentally uncertain phenomenon within a GIS, and is simply not repeatable across a spectrum of systems. This should not be seen as a case of algorithm or implementation *error*, since all may be assumed to be justifiable, but rather a case of algorithm *uncertainty*. If the viewshed operation is not precise, the Boolean representation and any logical combinations which may be made are not acceptable. Therefore, an alternative model for the viewshed must be sought, and the probable viewshed, which can be found by Monte Carlo simulation of elevation errors (but other methods are suggested by Teng and Davis 1992), is suggested as such a model here.

This work implies no criticism of any particular GIS implementation of the viewshed operation, since all are close to the range of values of the control analyses, and none is shown to be grossly wrong. The error is a conceptual one, and is in the precision of the viewshed implied by all operators. It is believed that the results obtained here would not have differed if the DEMs had been of any natural and variable terrain on earth, and the viewing locations anywhere within that terrain (except if it were at the bottom of a hole!).

9. Invitation

Only seven GIS packages were used in this study, and those are dominantly in the academic domain. If any other GIS developers would like to receive the test data, and run their own viewshed operations on the data, the author is very willing to supply the test datasets. Anyone participating should be able to return the resulting viewshed in ASCII grid format. If enough other implementations are forthcoming, the author will undertake to submit updated versions of table 5 and figure 8 for publication as a note in a future issue of this journal.

Acknowledgments

This work would not have been possible without the assistance and encouragement of many individuals. The following either provided software, or ran the experimental data through particular systems, I wish to thank them all for their time: Andy Baffes, Bill Carstensen, Bill Hickin, Ravi Narasimhan, Howard Veregin, and Jim Westervelt. Among others, Nick Chrisman, Jack Dangermond, Armondo Guevera, Robert Mills, and Dana Tomlin showed interest or provided encouragement in the research by word or by letter. Kate Moore assisted with figure preparation. Finally one reviewer provided very useful and perceptive comments on the original manuscript.

References

BURROUGH, P. A., 1986, *Principles of Geographical Information Systems for Resource Evaluation* (Oxford: University Press)

CLARKE, K. C., 1985, A comparative analysis of polygon to raster interpolation methods. *Photogrammetric Engineering and Remote Sensing*, **51**, 575–782.

DCDSTF (DIGITAL CARTOGRAPHIC DATA STANDARDS TASK FORCE), 1988, The Proposed Standard for Digital Cartographic Data. *The American Cartographer*, **15**, 11–142.

FELLEMAN, J., and GRIFFIN, C., 1990, *The Role of Error in GIS-based Viewshed Determination: A Problem Analysis.* IEPP Report No. EIPP-90-2, Syracuse, New York.

FISHER, P. F., 1991, First experiments in viewshed uncertainity: the accuracy of the viewshed area. *Photogrammetric Engineering and Remote Sensing*, **57**, 1321–1327.

FISHER, P. F., 1992, First experiments in viewshed uncertainty: simulating the fuzzy viewshed. *Photogrammetric Engineering and Remote Sensing*, **58**, 345–352.

HEMENWAY, D., 1992, GIS observer: benchmarks and comparisons. *Photogrammetric Engineering and Remote Sensing*, **58**, 31.

JORDAN, L. E., and STAR, J., 1992, A call for action: Standards for the GIS community. *Photogrammetric Engineering and Remote Sensing*, **58**, 863–864.

KNAAP, W. G. M. VAN DER, 1992, The vector to raster conversion: (mis)use in geographical information systems. *International Journal of Geographical Information Systems*, **6**, 159–170.

KVAMME, K., 1990, GIS algorithms and their effects on regional archaeological analysis. In *Interpreting Space: GIS and Archaeology* edited by K. M. S. Allen, S. W. Green and E. B. W. Zubrow (Basingstoke: Taylor & Francis), 112–125.

LEE, J., 1991, Comparison of existing methods for building triangular irregular network models of terrain from grid digital elevation models. *International Journal of Geographical Information Systems*, **5**, 267–285.

MARK, D. M., 1987, Recursive algorithms for the analysis and display of digital elevation data. *Proceedings of the First Latin American Conference on Computers in Cartography, San José*, Costa Rica (Columbus, Ohio: International Geographic Union Commission on Spatial Data Sensing and Processing), pp. 375–397.

PIWOWAR, D. J., ELLSWORTH, F. L., and DUDYCHA, D. J., 1990, Integration of spatial data in vector and raster formats in a geographical information system environment. *International Journal of Geographical Information Systems*, **4**, 429–444.

SRINIVASAN, R., and ENGEL, B. A., 1991, Effect of slope prediction methods on slope and erosion estimates. *American Society of Agricultural Engineers, Applied Engineering in Agriculture*, **7**, 779–783.

SKIDMORE, A. K., 1989, A comparison of techniques for calculating gradient and aspect from a gridded digital elevation model. *International Journal of Geographical Information Systems*, **3**, 323–334.

SKIDMORE, A. K., Terrain position as mapped from a gridded digital elevation model. *International Journal of Geographical Information Systems*, **4**, 33–49.

TENG, Y. A., and DAVIS, L. S., 1992, *Visibility Analysis on Digital Terrain Models and its Parallel Implementation.* Technical Report CAR-TR-625, Centre for Automation Research, University of Maryland.

TOMLIN, C. D., 1991, *Cartographic Modelling and Geographic Information Systems.* (Englewood Cliffs, N. J.: Prentice Hall).

TRAVIS, M. R., ELSNER, G. H., IVERSON, W. D., and JOHNSON, C. G., 1975, *VIEWIT: Computation of Seen Areas, Slope, and Aspect for Land-use Planning.* USDA Forest Service General Technical Report PSW-11/1975, Pacific Southwest Forest and Range Experimental Station, Berkeley, California.

WAGNER, D. F., 1989, *A comparison and evaluation of three polygon overlay algorithms.* Technical Report Series TR 89-01, GIS Laboratory, The Ohio State University.

YOELI, P., 1985, The making of intervisibility maps with computer and plotter. *Cartographica*, **22,** 88–103.