PUSHBROOM ALGORITHMS FOR CALCULATING DISTANCES IN RASTER GRIDS

J. Ronald Eastman
Graduate School of Geography
Clark University
Worcester, MA 01610, USA

ABSTRACT

Distance and proximity are critical variables in many geographic analyses. In raster geographic analysis systems, distance is most commonly determined by a sequential growth process whereby distances are accumulated in radial bands from an initial set of features. While such procedures are very efficient for the generation of small buffer zones, they become cumbersome when large distance surfaces need to be determined. As an alternative, two "pushbroom" algorithms are presented -- one for the case of calculating true Euclidian distance over a plane, and a second for incorporating frictional effects in the generation of cost distance surfaces. In the former case, a complete surface of any size can be calculated in exactly four passes through the data. In the second, as few as two complete passes are required, depending upon the nature of the frictional effects encountered. This economy arises from the nature of pushbroom techniques, whereby computations proceed sequentially (not radially) through the raster grid, acquiring directionally-oriented knowledge in accordance with the direction of the pushbroom path.

INTRODUCTION

A common requirement of raster-based Geographic Information Systems is the determination of distance. For example, if the distance from each grid cell to the nearest designated feature can be calculated, a buffer zone of any given distance may then be established. Buffer zones are essential planning tools in the exclusion or confinement of planning activities or investigations. A knowledge of distance is also essential when resources are clustered, and the type or level of activity that may be maintained is consequently distance-dependent. For example, the difference between animal species in the importance of distance to the nearest well is an important consideration in range management (Olsson, 1985, 81). Likewise, when resources are dispersed but access is limited, proximity to access points, such as roads, is an important management concern. Indeed, distance is a common ingredient in the assessment of processes that exhibit distance decay, including processes of mineralization, locational economics, and assessments of risk.

DISTANCE

Two broad approaches to the calculation of distance in raster-based systems are commonly in use. The first, and

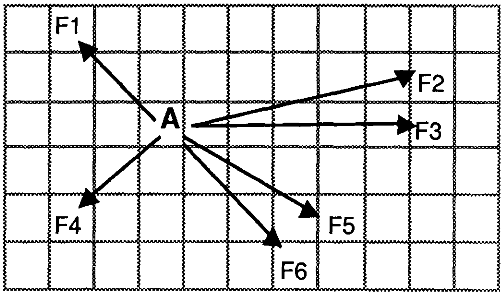

Figure 1 : In the most direct of the distance techniques, distance is calculated as the least distance between a cell and each of the designated feature cells. For example, the distance recorded at position A will be the minimum of the distances between A and each of the feature cells at F1 through F6.

most direct, relies upon simple Pythagorean geometry (Figure 1). Here the distance of each cell to the nearest of a set of designated feature cells is determined by calculating the Euclidian distance from that cell to each feature cell using row and column subscripts (eg., Olsson, 1985, 81). When distances from a single feature cell are required, the technique is quite efficient, with the number of operations being proportional to the square of the maximum distance (in grid cell units) required. In addition, the distances calculated are truely Euclidian. However, as the number of designated feature cells (ie., cells from which distance must be calculated) increases beyond one, the procedure requires that the nearest neighboring feature cell be determined for each cell in the grid. As a result, efficiency is proportional to the number of feature cells as well as the maximum distance involved. In addition, the positions of all cells belonging to the designated features must be known in advance (or be determined from the raster), with their coordinate positions being held in some form of accessible stack or array. Given the complexity of many GIS feature patterns, the technique can thus quickly become bogged down.

To avoid these problems, a second approach (eg., Tomlin, 1986) employs the concept of growth rings. Initially, each of the feature cells is tagged with a distance of 0 while all other cells are marked with a distance equal to the maximum distance that will be determined. Then in a series of passes through the image, distance is "grown" from the feature cells in a series of concentric rectangular "rings" until the maximum distance is reached (Figure 2). The technique has the very strong advantage that the locations of the feature cells do not need to be stored in an accessible list, nor do any nearest-neighbor calculations need to be made. By definition, each growth ring will be constructed with reference to the nearest feature cell, and distance to that feature cell can always be determined by examining the squared distance of an adjacent cell within the previous growth ring. Specifically, squared distance is a linear combination of squared distance in X and squared distance in Y (the Pythagorean theorum). As a result, squared distance is also equal to that of any intermediate distance plus the differences in squared X and squared Y between the new point and that intermediate. Each ring is therefore grown by adding onto the distance of the edge cells the difference in squared X and squared Y. As it turns out, differences in squared X and squared Y systematically increase by an increment of 2 (Figure 3). The necessary increment can therefore be determined by looking at the previous increment in that direction and adding 2.

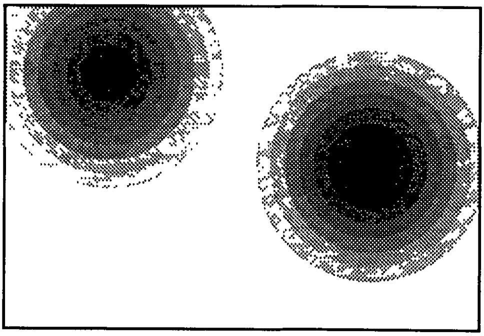

Figure 2 : In a second approach, distance is "grown" in concentric rings around each feature. The shaded bands here indicate the growth ring stages in this process. If the process is continued far enough these rings will coalesce to form a continuous distance surface. Like the method indicated in Figure 1, distances are Euclidian.

	B									
	A	18	13	10	9	10	13	18		
		13	8	5	4	5	8	13		
		10	5	2	1	2	5	10		
		9	4	1	0	1	4	9		
		10	5	2	1	2	5	10		
		13	8	5	4	5	8	13		
		18	13	10	9	10	13	18		

Figure 3 : With distances being stored as squared distances, new distances can be determined by adding incremental changes in delta X squared and delta Y squared. For example, Cell A differs from the cell in the upper-left corner by 0 in squared Y and 7 (ie. [18-13]+2) in squared X. The distance of Cell A is thus equal to 18+0+7=25. Incremental squared distances always differ by 2. Thus the distance of cell B is 18+7+7=32 -- i.e., the previous difference of 5 in squared X plus 2 plus the previous difference of 5 in squared Y plus 2.

The use of squared distance has several advantages. In addition to being required by the algorithm, the avoidance of square roots significantly speeds operations. Only after all growth rings have been calculated would a final pass be made to take the square roots of cell values. In addition, the intermediate storage of squared distances allows perfect precision with integer data. As a result, rounding errors do not accumulate but only affect the results of the final pass. The procedure, then, does have a considerable amount of appeal. However, it is also not without some problems. First, if integer arithmetic is to be used, most applications will require 32 bit integers because of the need to store squared distances (with 16 bit integers, the maximum distance that may be calculated is a mere 181 cells). Second, the procedure requires the ability to move quite freely about the image cells. Random access is trivial if the entire image is in memory, but with 4 byte integers or floating point values, the size of image that will readily fit into memory may be quite limited. Random file access can alleviate this, but the speed of random disk operations is typically quite slow. Finally, and perhaps most significantly, the number of passes that must be made through the image is a direct function of the maximum distance that must be calculated. While the determination of narrow buffer zones will be quite efficient, the calculation of a continuous distance surface over any extensive region would likely be quite slow.

COST DISTANCE

An interesting feature of the "growth" procedure is that its logic may also be developed to incorporate frictional effects. Whenever distance is used to imply the cost of movement, that cost will be a function not only of distance, but also of the frictional effects of various relative and absolute barriers such as land cover and slope. This new measure may be called "cost distance", and may be evaluated in any meaningful unit involving distance, money or time.

In the evaluation of cost distance using a growth process (Tomlin, 1986), a matrix is first constructed containing the designated feature cells marked with a distance of 0, and with all other cells being tagged as unknown. In addition, a second matrix is formed in which the frictional effect of each cell is stored. All frictions are indicated with a value relative to 1. Thus, for example, a friction of 2 would indicate that it costs twice as much as usual to pass through that cell. The procedure then involves a series of passes through the matrix in which unknown cells which are adjacent to a cell of known distance are given a distance equal to the known cell plus one times the frictional effect in the cardinal directions and a distance equal to the known cell plus 1.41 (square root of 2) times the frictional effect in the diagonal directions (Figure 4).

Feature image

????	????	????
????	0	????
????	????	????

+

Friction image

1.00	1.00	1.00
1.00	1.00	2.00
2.00	3.00	3.00

=

Cost distance

1.41	1.00	1.41
1.00	0.00	2.00
2.82	3.00	4.23

Figure 4 : Distances grown outwards from any feature cell would normally result in an increase of one in the cardinal directions and 1.41 along the diagonals. However, frictional values other than 1.00 will proportionately alter this relationship.

Unlike the simple distance growth model, the concentric "rings" are no longer rectangular in shape, but octagonal. In addition, distances are accumulated directly, rather than as squared distances. The reason for this relates to the fact that there is no longer any predictable relationship between distance and the difference in X or Y between a grid cell and its nearest target (because of the variable effects of friction). Errors will therefore accumulate for any cost distances determined along paths other than one of the cardinal directions or the diagonals. As a result, most systems, such as IDRISI (Eastman, 1987) and the Map Analysis Package (Tomlin, 1986) provide both a simple distance routine as well as one for calculating cost.

Given its inherent "growth" logic the cost distance routine discussed above shares most of the same strengths and weaknesses as the simple distance growth routine. Again, the procedure is very efficient whenever cost needs to be determined over a narrow buffer zone. However, it likewise bogs down whenever a significant region must be determined. For example, to calculate a continuous cost surface over a 512 by 512 grid could involve over 700 passes through the data set to construct the required number of growth rings. Similarly, the need for random access can cause a tradeoff between image size and speed.

THE IDRISI SYSTEM APPROACH

During the development of the IDRISI GIS system, new procedures for the calculation of distance and cost surfaces were developed. The IDRISI system is a grid-based (or "raster") geographic analysis system that has been developed by the author at Clark University. It is also distributed by the university with over 600 registered users at this time.

The IDRISI system was specifically designed to operate in a microcomputer environment in which disk space is plentiful (eg. 32 Mb), but random access memory is scarce (640 Kb). As a result, all procedures were developed using a scan line approach whereby only a limited number of scan lines would be operated upon at one time. The procedures developed for the calculation of distance and cost thus follow a scan-line approach whereby successive rows of the image are read and operated upon, and then saved back to disk. In both cases, the procedures operate by pushing effects through the image, much like a pushbroom would be used to systematically clean a room. Effects then ripple through the image, much like water being pushed over a wet floor.

THE PUSHBROOM DISTANCE PROCEDURE

In the case of the simple Euclidian distance algorithm (called "DISTANCE" in IDRISI), processing starts from the upper-left cell and proceeds along each row and then sequentially down the image from one row to the next. This is identical to the order in which the image is stored. As the feature image (from which distances must be calculated) is read, a temporary data file is output with records which record the distance in X and the distance in Y (as floating point real numbers) to the nearest target cell that is either above or behind it. If no feature has yet been found, these values are output with a special flag value. However, once a feature is found, distance in X and Y are carried along by incrementing by one for each successive column or row. When more than one feature has been found, delta X and delta Y are recorded from the nearest one by comparing distances determined from incrementing the delta X and delta Y values from the cell to the left, the cell diagonally above, and the cell directly above. In essence, the effect is one of determining the lower-right quadrant of all growth rings in a single pass --a kind of ripple effect in which knowledge of feature positions is carried along in the pass (Figure 5).

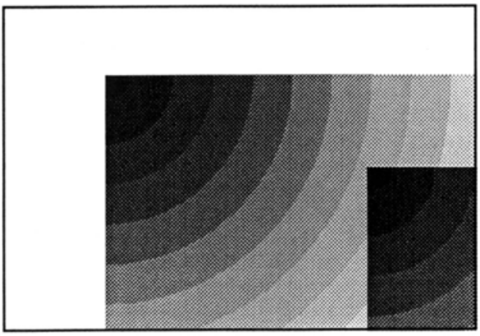

Figure 5 : During each pass of the pushbroom procedures, effects are "pushed" from regions already examined (in this example, from cells above and behind) into regions yet to be processed. The procedure thus attains directionally-oriented knowledge with each pass.

This procedure is then repeated, three further times. Whereas the first pass proceeded from top-left to bottom-right, successive passes then proceed from bottom-right to top-left (to determine the upper-left quadrant of the growth rings), from top-right to bottom-left (to determine the lower-left quadrant) and finally from bottom-left to top-right (to determine the upper-right quadrant). These four passes are then overlaid with the final output being stored as the minimum distance calculated from the delta X and delta Y figures for each pass.

In its implementation, several economies can be used. First, as each row is processed, only that row and its immediately preceeding row need be held in memory. By using scan-line buffers in memory, the beginning of any required row (regardless of the direction in which it would be processed) would be randomly accessed, with all remaining row values being read sequentially in normal file order. Second, the overlay step does not need to be done at the end, but can be done on a row by row basis during each pass. All that is required is that delta X and delta Y be recorded in a consistent coordinate system with negative values in the left and bottom quadrants. In this way, the results stored after each pass represent the best estimates of least-distance delta X and delta Y for all passes up to that point. Finally, all intermediate distance calculations can be made using squared distance. Only on the last pass does the square root need to be taken.

As a result of this procedure, distance can be calculated as a continuous surface in four passes regardless of the size of the image, the number of feature cells, or the maximum distance required. That said, it does require that the full surface be calculated every time. To create a buffer zone, then, requires that the distance surface be reclassified into cells within the zone distance and those outside it. One might expect, then, that the procedure would be slower than the traditional growth ring approach for narrow buffer zones but faster whenever a more significant region must be defined.

THE PUSHBROOM COST PROCEDURE

In the case of the cost distance routine (called "COST" in IDRISI) a somewhat similar procedure is used. Again, sequential passes are made through the data, but as few as two complete cycles are required depending upon the nature of the frictional effects involved. First, however, the issue of friction needs to be discussed.

Frictional effects present barriers that are either absolute or relative in nature. An absolute barrier is one in which the frictional effects are so high that movement cannot proceed through that cell. Relative barriers, however, do allow movement, albeit at an additional cost. Some systems treat absolute barriers as special cases. However, in IDRISI, an absolute barrier is indicated simply by giving that cell a friction that is impossibly high (ie., one that will always cause distance to be shorter by moving around the barrier than over it).

Given this concept of friction, the cost distance procedure processes the file from top-left to bottom-right and then backwards from bottom-right to top-left. During the first pass it sets all unknown cells to have an extremely high distance, and all feature cells to have a distance of zero. Additionally, like all subsequent passes, it examines the 8 neighbors about each cell to see if distance incremented from that neighbor is less than the distance currently stored for that cell. Like the growth procedure for calculating cost, distance is incremented as one times the friction in the cardinal directions and 1.41 (square root of two) times the friction along the diagonals.

As long as there are no absolute barriers (ie., as long as going over a feature is always less expensive than going around it), the complete cost surface can be determined in two full passes from the position of the first feature. For example, if the first feature cell is found half-way through the image, the procedure would minimally require the first pass down the image (in which the first feature is found), the second pass back up the image, and a third pass back down the image until the position of the first feature cell is found again. When absolute barriers are present, however, their nature and position may disturb this rule. For example the barrier in Figure 6 Part A, causes no problems since information from the feature is carried to all parts of the image. In Figure 6 Part B, however, the position of the barrier prevents information about the location of the feature from being carried to the bottom right-hand corner on the first pass. Unless a complete third pass is undertaken, distances will not be correct in the region indicated. Generally, the procedure will have difficulty with absolute barriers that produce maze-like corridors. However, for natural resource applications where relative barriers predominate and absolute barriers are not complex, three complete passes have generally been found to be adequate.

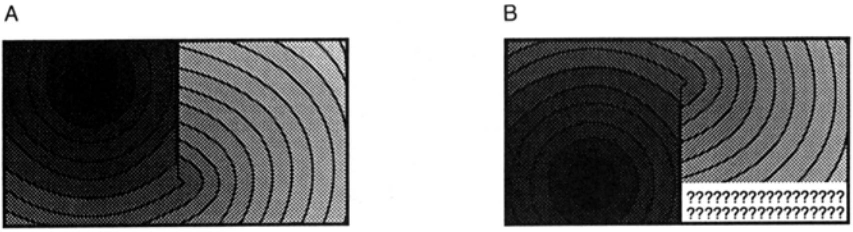

Figure 6 : Absolute barriers may necessitate more passes than the minimum required for a surface with only relative barriers. The absolute barrier in A, for example, poses no problems for the routine since the top-left to bottom-right and vice-versa pass orientation will carry information about the position of that feature to all cells in the image. The barrier in B, however, will block the complete spread of information unless at least three complete passes are used.

As with the pushbroom procedure for calculating simple distance, the pushbroom cost procedure produces an algorithm where the number of passes is independent of the maximum cost distance to be determined. Similarly, since the entire image is processed, the procedure is quite efficient in instances where a complete cost surface is required.

A COMPARISON OF TECHNIQUES

A comparison of the pushbroom and growth ring techniques is difficult to evaluate. The strengths of one are the weakness of the other. For the determination of small buffer zones, there is little doubt that the growth ring procedures will be superior, since their speed is directly related to the size of the buffer required. However, whenever a full distance surface is required (cost or simple Euclidian), the pushbroom techniques should be faster. To evaluate this, the two techniques were compared by applying them to identical full-surface problems. The first test (to be called the "center" test) involved a single-cell feature in the center of the image while the second (to be called the "corner" test) involved an image with single-cell features in each of the four corners of the image. Growth procedures typically limit their operation to the maximum subregion required for processing during any one cycle. As a result, operations should be fastest for the "center" test and worst for the "corner" test. For the pushbroom procedures, however, the results of these two tests will be identical. These two tests where then applied to both the simple Euclidian and cost distance problems for images which ranged in size from 25 x 25 cells to 175 x 175 cells. For all cost distance tests, frictions were set at 1.0 for all cells.

For the growth procedures, the "SPREAD" routine from the microcomputer version of the Map Analysis Package

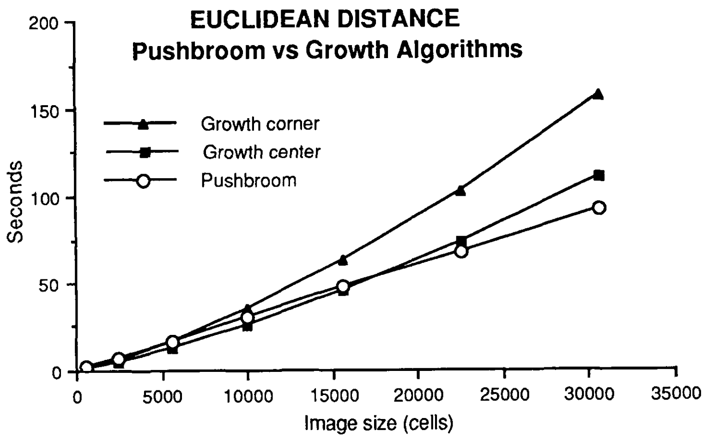

Figure 7

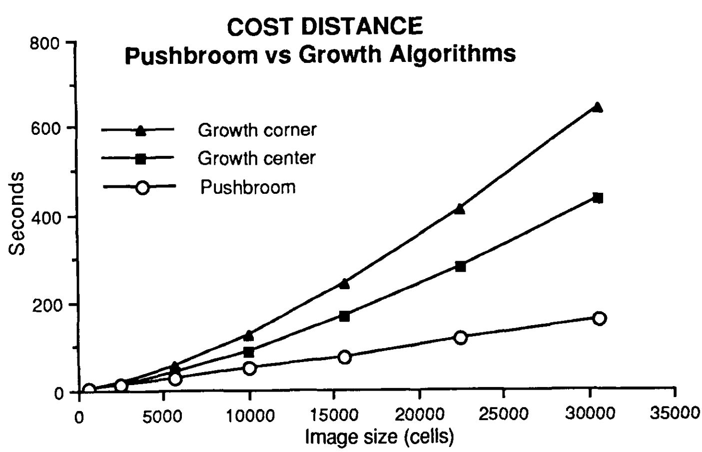

Figure 8

(Tomlin, 1986) was used. This routine incorporates both the simple distance and cost distance growth routines (which it switches between depending upon whether one specifies a friction surface to be spread "through"). For the pushbroom routines, the "DISTANCE" and "COST" modules from the math coprocessor version of IDRISI were used (a version still under development as of this writing). Both systems were run on a 20 Mhz 80386 computer with a 128 Kb disk cache in use. In addition, both programs accessed the 20 Mhz 80387 coprocessor installed on the test machine. Figures 7 and 8 present the results of these two tests.

As can be seen from these results, the times for the pushbroom algorithms are a linear function of image size. For the growth routines, however, processing time can be seen to be exponentially related to image size with times for the corner test being, as expected, uniformly greater than for the center test. The images used in these tests were not large, and yet the savings in processing time afforded by the pushbroom techniques are immediately apparent -- particularly for the cost distance tests. For example, extrapolating these results to a 1024 x 1024 image (this test could not be done with the Map Analysis Package since its maximum image size is 32,640 cells [180 x 180]), the pushbroom cost procedure would require 1.5 hours while the growth ring procedure would require almost 11.5 hours for the center test and 20 hours for the corner test!

CONCLUSIONS

From the above, several broad conclusions can be reached about the techniques introduced in this paper. First, the pushbroom procedures provide a logic compatible with scan-line processing. As a result, they may be applied to very large images even though memory may be limited (such as the 640 Kb address space of MS-DOS). Second, like growth ring procedures, they do not require explicit information about the location of the features from which distance is to be determined. Third, their speed is a linear function of image size. Fourth, they operate upon the entire image, thus making them somewhat inefficient for determination of small buffer zones (for which the growth ring procedures excel). But finally, they are substantially faster than the growth ring processes whenever a more substantial buffer zone must be processed or a continuous distance or cost surface must be determined.

REFERENCES

Eastman, J.R., (1987) IDRISI, A Grid-Based Geographic Analysis System, Graduate School of Geography, Clark University, Worcester, MA, USA.

Olsson, L. (1985) An Integrated Study of Desertification: Lund Studies in Geography, Ser.C, No.13.

Tomlin, C.D., (1986) The IBM Personal Computer Version of the Map Analysis Package, The Laboratory for Computer Graphics and Spatial Analysis, Harvard University, Cambridge, MA, USA.