A Proposed Standard Test Procedure

for Conversion Between Raster and Vector Data

Joseph M. Piwowar
Ellsworth F. LeDrew

Department of Geography
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada

ABSTRACT

The present trend in Geographic Information System (GIS) design is towards the incorporation of more diverse spatial data sets. Analysts often wish to combine remotely sensed images with more conventional data (soil types or political jurisdictions, for example) which are mapped using lines and polygons. The difficulty in integrating gridded (raster) imagery with (vector) polygons can be a major obstacle in the data analysis. There have been several studies published which have attempted to identify the best algorithm to use when converting data between vector and raster formats. All of these reports, however, have based their conclusions on theoretical execution times. In this paper, we propose a more comprehensive evaluation procedure which includes measures of the quality and accuracy of the conversion results as well as algorithm efficiency. These tests should be performed using data sets which have been specifically designed to fully exercise the capabilities of each algorithm as well as applied to files which are representative of real-world applications.

INTRODUCTION

Since the launch of Landsat 1, the science of remote sensing has flourished and there have been many published studies documenting new techniques and procedures for image analysis. We are quickly realizing, however, that there is a much greater potential in the utility of remotely sensed data if it can be combined with other spatial data sources in a geographic information system (GIS).

Remotely sensed imagery is recorded in a *raster* (grid) format. Most conventional data sources, on the other hand, are digitized as a sequence of lines and polygons, that is, in a *vector* form. The difficulty which any GIS must overcome is that there is much information recorded in either vector or raster format, but these are mutually incompatible (van Roessel and Fosnight, 1985). Many existing GISs address this difficulty by requiring that all data be stored in one form or the other. This means that there must be data conversion from raster format to vector, or *vice versa*, before processing can begin.

Conversion between vector and raster formats is not a new research topic - it has been discussed for over two decades in the computer graphics literature. In a computer graphics display, each point can be simply drawn without regard to its spatial connections with the other information in the image. Converting data for GIS applications, however, is a much more difficult task since we need to retain the spatial relationships inherent in the data structures. For example, which polygons are islands contained within larger polygons?

Previous researchers (Franklin, 1979; Pavlidis, 1979; Ackland and Weste, 1981; Peuquet, 1981a and 1981b) have addressed the data conversion problem by theoretically evaluating the methods of converting spatial data files between vector and raster formats and have based their conclusions on the techniques which potentially have the shortest execution times. The results of these studies are often inadequate to predict how an algorithm will react to real-world data. There is too much emphasis placed on execution times and not enough attention given to the accuracy and quality of the results. A program which executes quickly yet produces erroneous results is of little use; one which generates data of high quality is always a better choice, no matter how long it takes to run.

In this study we build on the past studies and describe a complete testing environment which can identify potential problems in a procedure before it is used operationally. The testing environment will be described in three sections: test data sets, test criteria, and test procedure.

TEST DATA SETS

In order to observe the operational characteristics of a conversion algorithm under a variety of conditions, it should be tested using *both* real and synthetic data. The data in the real test files should closely match the types of data which the procedure will be expected to handle. During this study, we used sample data from a manually digitized geologic map for evaluating vector to raster conversions and a classified Landsat Thematic Mapper sub-scene to test the raster to vector conversion algorithms (Figure 1). The use of real-world data is important since it is impossible to foresee all of the potential data complexities using simplified data sets alone.

We have designed a synthetic test data set (Figure 1) to tax the capabilities of an algorithm in several critical areas. The quality criteria listed below provided the primary rationale for the design of these files, so these rationale will be discussed in their respective criteria sections.

TEST CRITERIA

In order to obtain an unbiased evaluation of the results of a conversion between vector and raster data, they must be subjected to a wide variety of tests. From an examination of previous studies (Muller, 1977; Carlbom and Michener, 1983; and Selden, 1985) and from insights gained by programming and working with many conversion algorithms, we have developed a group of test criteria which have been proven to be very effective in measuring the performance of these conversions. As outlined in Table 1, the test criteria can be grouped into quality, accuracy and efficiency categories. We will devote most of our attention in this paper to the discussion of the qualitative criteria since they are spe-

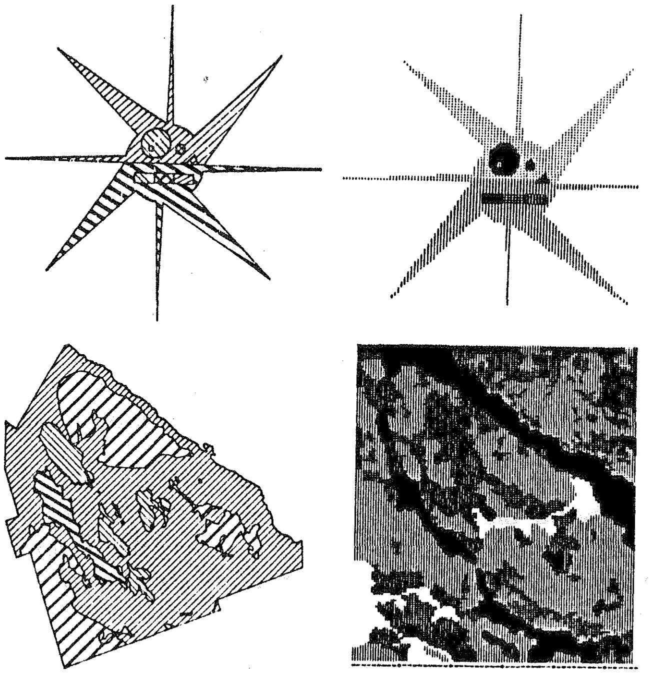


Figure 1: Test Data Sets (Vector and Raster Forms). top: synthetic image; bottom: real data from the Chalk River Properties of Atomic Energy of Canada Limited.

cific to vector to raster and raster to vector conversions. These quality measures will be presented separately for each conversion direction. The accuracy and efficiency criteria are generally well understood so only an overview of these categories will be presented.

Quality Criteria for Vector to Raster Conversions

Polygon Fill: Once the polygonal boundaries have been transferred to raster form, the interior of the regions must be filled with the appropriate class code. There are many approaches to polygon fill and their effectiveness is generally proportional to the amount of the image that can be accessed at one time. Those techniques which can examine the entire grid simultaneously usually produce better results than those which have access to one or two lines. Even the best filling algorithms, however, are not without limitations. These techniques are susceptible to the size of the image pixels relative to the data complexity. For example, it is not unusual in remotely sensed imagery to have long, narrow features, such as rivers, having widths varying from one or two pixels to several cells. The difficulty in unaided filling of these regions is a common cause for many of the other errors experienced in vector to raster conversions.

Problems during the filling stages of the pixel regions are most often observed visually. The synthetic test object can be used to test the accuracy of the filling of the interior of the rasterized polygons. The single-celled extremities at the spine tips and

Table 1: Evaluation Criteria

1. Quality Criteria for Vector to Raster Conversions
   a. Polygon Fill
   b. Cut Cell Assignment
   c. Cell Size Selection

2. Quality Criteria for Raster to Vector Conversions
   a. Line Aliasing
   b. Island Detection
   c. Conversion of Diagonally Connected Pixels
   d. Object Identification

3. Accuracy Criteria
   a. Areal Accuracy
   b. Perimeter Accuracy
   c. Displacement

4. Efficiency Criteria
   a. Memory Efficiency
   b. Disk Efficiency
   c. Time Efficiency

the horizontal class boundary across the centre of the object may uncover some difficulties in this area.

Cut Cells: It would be very unusual to have a polygonal image whose edges displayed perfect orthogonality at a constant interval. If such a data set was encountered, then it could be converted to a raster form unambiguously. More realistic polygons have boundaries which cannot be mirrored exactly by square pixels. These edges tend to intersect and cut across the image grid hence these pixels are called cut cells. The question which the program must address is: "To which polygon is a cut cell assigned?"

There are three basic approaches to handling cut cells: assign them to the region which has the largest fraction of the cell's area (*dominant class rule*, Nagy and Wagle, 1979); assign them to the region which has lost the most area during previous cut cell assignments (*surface balance rule*, Bonfatti and Tiberio, 1983); and ignore them. In the dominant class approach, the resulting raster array has a high accuracy on a per pixel basis, but may have significantly high area measurement errors over the entire scene. The surface balance technique has exactly the opposite effect. Unfortunately, most vector to raster conversion algorithms follow the third approach which may produce unexpected spatial coverage errors. The scale of the spatial errors can be kept to a minimum if the pixel size is small relative to the data's complexity.

Most commercially available conversion algorithms do not specify which approach to cut cell assignment they use (which usually implies the "do nothing" technique). The performance of a particular procedure is difficult to find during a visual evaluation of a converted polygon file, but can often be noted by looking for changes in area accuracy.

Cut cells will be found throughout the synthetic image, but of particular interest is the examination of the pixel values assigned near the tips of the spines and in the core area, where the potential for incorrect assignment is high.

Cell Size: As alluded to in the previous two criteria, the selection of pixel size in the raster image can have a critical effect on the degree of satisfaction with the results of a vector to raster conversion. If the pixel size is selected to be too small, an unmanageable quantity of data would quickly result. If the cells were too big, however, some of the smaller features in the image array become lost during the conversion. The optimum pixel size is dependent on the size and complexity of the polygons to be converted: a good choice might be one quarter of the size of the smallest feature (Hanley, 1982). Unfortunately, this may be difficult since the grid may be required to have a specific dimension in order to be compatible with other data sets in the GIS.

The tapered spines protruding from the core of the synthetic image can be used to illustrate the effects of using different cell sizes by examining the extent to which they are generalized in the converted polygons. In addition, the core of the object is populated with island objects of a variety of sizes: their presence or absence from the raster approximation could be noted. A visual examination of the raster image produced by the conversion algorithm will give a good indication of how the pixel size has affected small regions.

Quality Criteria for Raster to Vector Conversion

Aliasing: Aliasing is an undesirable artifact of the square shape of the fundamental raster unit, the pixel. As a region boundary is extracted from an image, the natural tendency to copy it as precisely as possible gives these edges a stepped appearance. A good conversion algorithm will minimize the aliasing effect during the conversion process. Most procedures, however, do not make any attempts to reduce the stepped appearance and require that a post-conversion anti-aliasing (boundary smoothing) program be applied.

One side of each spine protruding from the core of the synthetic image has been drawn at a precise increment of forty-five degrees, while the other sides represent edges at arbitrary angular increments. This arrangement highlights the aliasing effects of converted edges at a variety of orientations. Aliasing will be evident during a visual examination of the conversion results. More quantifiable measures can be obtained by comparing the perimeter distances around each pixel region before and after conversion and by noting the amount of disk space required to store the vector data set. These last two measures, however, can be affected by factors other than aliasing.

Islands: In order to generate topologically accurate vector files, it is not sufficient to simply record the segment coordinates defining the *outside* edge of a polygon. The polygon definition must also include a list of any other regions which are islands (polygons totally contained) within its bounds. These islands form an *inside* edge to the polygon which must be taken into account when displaying it or calculating its area. A good raster to vector conversion algorithm must be able to identify one or more islands within a region, and do this to any depth (i.e. flag islands nested within islands). While many techniques can do this successfully for *simple islands* (isolated polygons within a larger region), they often fail to locate *island clusters* (several adjacent polygons which are all islands in a larger region).

To test for island identification, compare the area of a pixel region containing islands (both simple and clustered) before and after conversion. A visual examination of a shaded plot of the parent polygon and its islands will also reveal which inner outlines have not been identified as islands since the outer region's shading will be drawn over the inner outlines. The synthetic test image has simple islands, nested islands, and island clusters which can be used to evaluate an algorithm's island detection capabilities.

Diagonally Connected Pixels: Due to the boundary tracing technique used, some algorithms cannot correctly handle pixels defining the same region which are only joined at a corner. Diagonally connected pixels should ideally be smoothly incorporated into the same outline. Many algorithms, however, would split the polygons at this point and make them distinct. Not only does this break continuity in the vector representation, but the extra polygons created increase the processing overhead associated with the data set. If there are many long and narrow linear features in the image then this can be a major concern.

A comparison (either visual or numerical) of the number of polygons created for a long diagonal feature in the image will reveal any problems encountered during conversion. The spines of the synthetic image drawn in the four cardinal directions intentionally have a very narrow taper to them. Any truncation of the length of these spines in the vectorized image indicates that the procedure has difficulty converting diagonally connected pixels.

Object Identification: The fourth issue which may pose some difficulty to those routines that process the raster image sequentially from top to bottom (or bottom to top) in scan line strips is that of correct object, identification. Suppose we were processing a "U"-shaped feature sequentially from top to bottom. When we first encounter the region we cannot determine if the left branch of the "U" is joined to the right branch, so we must assume that they are unique regions. As such, all segments and vertices around the left branch are given different polygon identifiers from those elements around the right branch. When we reach the bottom, however, we realize that the left and right branches are from the same polygon and should be merged with a common identifier. An accurate conversion algorithm must, at this point, re-trace the boundary of either of the branches and reset the polygon identifiers for any segments, vertices and adjacent polygons which may be affected.

As the synthetic image is processed, multiple branches of the same object will be merged as the algorithm nears the core area and then split again as the procedure moves towards the image edge. This facilitates the monitoring of the results of each merge and split to ensure that the polygon identifiers are being correctly updated where necessary.

Accuracy Criteria

In many respects, accuracy is the quantification of quality. An algorithm which experiences errors in polygon fill will have inaccurate areal measurements. Three separate measures of accuracy can be used for either raster to vector or vector to raster conversions: area, perimeter and displacement.

Area Accuracy: An object and its converted approximation should cover approximately the same areal extent. A comparison can be made of the areas occupied by each object in the scene before and after conversion. For raster files, this is a simple pixel tally and for vector data a routine can be invoked which calculates the area under a curve.

Perimeter Accuracy: The perimeters of the pre- and post-conversion objects can be compared for significant differences. The perimeter of a polygon is calculated as the sum of the linear distances between sequential boundary vertices. Similarly for raster images, the perimeter is the total of the distances between the centres of the boundary pixels which are adjacent to an exterior region along a pixel side. Note that some researchers measure the perimeter of a pixel region by summing the lengths of all exposed pixel edges. Due to aliasing, this technique tends to give exaggerated results which are not directly comparable to polygon-based perimeter calculations, hence should not be used.

Displacement Accuracy: It is not uncommon, during a conversion operation, that the final scene gets shifted slightly. Displacements of one pixel width can be expected along the edges of polygons and pixel regions. These arise from the degradation of planimetric accuracy during vector to raster conversions. Non-systematic displacements, however, indicate problem areas in the conversion routine and should be flagged.

Displacements can be observed by double-converting each data file. That is, raster files can be converted to vector approximations and the resultant polygons are converted back to raster images. A similar approach is followed for the vector data sets. The displacement images are then plotted against the original data for a visual examination of the magnitude and distribution of any displacement.

Efficiency Criteria

Memory Efficiency: There is a general axiom for processing large data sets, such as those commonly encountered in a geographic information system: the more data that you can hold in memory without writing it to disk, the faster your program will run. Memory, or core storage, is the part of the computer where all data must reside immediately before and after being used in calculations. Since accessing a value which is currently in a memory location takes much less time that reading that same number from a disk file, those programs which store much of their data in memory tend to be faster. However, all computer systems have a physical limit to the amount of information that they can retain in memory. If a program exceeds that barrier, it may not run at all, or it may execute more slowly since the computer is then forced to start copying parts of its memory to disk for temporary storage.

A measurement of memory utilization during processing may not be a straightforward procedure and will most likely vary between computer systems. Consultation with a systems programmer and/or the system's manual set will describe how to measure computer storage utilization.

Disk Efficiency: The ultimate result of a conversion procedure is a disk file containing the converted scene. The quantity of data produced by different conversion programs can vary considerably. A procedure which generates larger files incurs a higher cost, not only in the initial writing and storage of that data, but also in the subsequent use of the files by other applications. A second factor to be considered is the amount of temporary disk storage required by the algorithm. If a program dumps part, or all, of its working storage to disk during execution, it could conceivably exhaust all of the available disk space which would result in the suspension of further processing. The fact that the storage required is only temporary - to be released upon successful completion - is immaterial in this situation.

Disk utilization can be measured by noting the size of the files created by the conversion. This may be misleading, however, since a particular file structure might have more operating system overhead than another. Consequently, disk efficiency can be more effectively quantified by measuring the amount of data created (e.g. number of vertices found).

Time Efficiency: An important criterion in any operational application is the length of time that it takes the process to complete. Using small data sets, a few seconds difference between the completion of a fast algorithm and a slow one may not be significant. However, these seconds can turn into minutes or hours when applied to large, complex files typical of a GIS. Two measures of time can be recorded: CPU time and elapsed time. CPU (central processor unit) time is a measure of the actual period that the computer spent solving this particular problem. It gives a good indication of the relative amount of computing power which is required to perform the task. Elapsed time is the total real time that the program takes to complete. This is the length of time that the operator must wait before being able to proceed. Two algorithms may have similar CPU times, but vastly different elapsed times if, for example, one does considerably more I/O (which is slow) than the other.

A useful calculation to perform is to compute the ratio of CPU to elapsed time for a conversion. A low ratio may be indicative of a procedure which is *I/O bound* (i.e. it spends much of its time reading and writing data to the disk). Such a procedure can be made more efficient by improving the disk interface. If the CPU to elapsed time ratio is high, then the program is busy computing most of the time. These procedures can be expected to be adversely affected when implemented on time-sharing computer systems where there is much contention for the CPU.

Unfortunately, many micro-computer operating systems do not provide tools for the measurement of CPU time. In these situations, an estimate of the elapsed time can still provide useful data for the overall evaluation of an algorithm's performance.

TEST PROCEDURE

Each selected algorithm should be evaluated by observing its performance characteristics with particular reference to the test criteria as it is used to convert the test data sets. Measurements can be made for those criteria where this was possible (e.g. time efficiency) and the quality parameters (e.g. displacement) can be subjectively evaluated. To account for any minor fluctuations in system performance, each test should be executed three times and the applicable results averaged.

SAMPLE ANALYSIS

To demonstrate the utility of the test procedure, it was applied to the comparison of three vector to raster conversion algorithms (Piwowar, 1987). The results are summarized in Table 2. We can see from this table that none of the algorithms were very good at polygon fill, with the Image Strip procedure providing unacceptable results. Cut cells were generally handled by the "do nothing" approach which gave variable results, depending on the cell size. The Scan Line algorithm received a higher ranking for this criterion since although it still did not consider the proportion of each region within a pixel, it did assign cut cells in a consistent manner (always to the polygon which filled the cell's bottom right corner). All of the algorithms reacted differently to changes in cell size (Figure 2), but each had acceptable results depending on the application. For example, the Scan Line algorithm will eliminate sub-pixel sized features from the conversions which gives it the best planimetric accuracy. The other two procedures preserved such features, which may be equally important from a contextual perspective.

Area accuracies varied from quite good, for the Scan Line algorithm, to average, for the Image Strip technique. The algorithms all had similar results for perimeter and displacement accuracies.

The rankings for computer memory utilization for each technique mirrored their requirements for random access to the image array: the greater the need to be able to look at any part of the image at once, the higher the the memory costs were. A general axiom for processing large spatial data sets holds that increasing the amount of data held im memory at one time will speed up execution times. While this is generally reflected in the results, an anomaly became apparent as the Image Strip algorithm ran faster using less memory than the Frame Buffer technique. A detailed analysis revealed that the Frame Buffer algorithm had been programmed to use an inefficient polygon fill procedure.

We can conclude, therefore, that the Scan Line algorithm clearly produced raster imagery with the best overall quality among those tested. While it was not as efficient as the Image Strip technique, its execution speed and disk utilization characteristics were still acceptable.

SUMMARY

In this report, we have presented an evaluation methodology for programs which convert spatial data between vector and raster forms. We have found that traditional approaches usually test only execution times and fall short of identifying the operational characteristics of a procedure in an actual GIS environment. In our approach, we have developed tests for a variety of criteria under the categories of quality, accuracy and efficiency. This methodology has been applied for several conversion techniques and has been proven to provide meaningful results. We are continuing our evaluation of different algorithms operating in both conversion directions using these criteria.

Table 2: Results of Benchmark Tests of Vector to Raster Conversion Algorithms on Real Test Data Sets

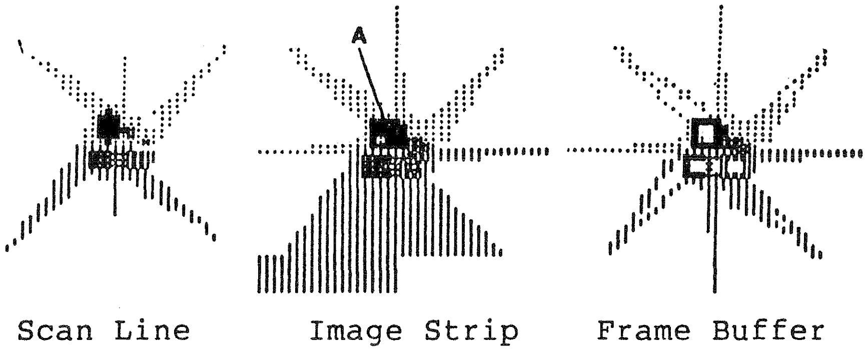
the best procedure for each test criterion is highlighted.

	Frame Buffer[1]	Image Strip[2]	Scan Line
Polygon Fill:[3]	C	D	**B**
Cut Cells:[3]	C	C	**B**
Cell Size:[3]	**A**	**A**	B
Area: (% change)[4]	5	9	**1**
Perimeter: (% change)[4]	3	**2**	4
Displacement:[3]	**B**	C	C
Memory: (page faults)	1066	1010	**972**
Disk: lines/pixels scratch	**404/426** **0**	**404/426** **0**	**404/426** **0**
Time: CPU/elapsed (m:s)	**0:17/0:30**	0:23/0:39	0:58/6:18

Notes:
1. Frame Buffer size = 1024 lines * 1024 pixels * 2 bytes
2. Strip size tested = 32 lines
3. A = no problem, B = above average, C = acceptable, D = error prone
4. % change is the average of the percentage differences measured for each class group before and after conversion

ACKNOWLEDGEMENTS

This research was supported by NSERC operating and infrastructure grants to Dr. E.F. Ledrew, and a contract with Atomic Energy of Canada Limited.



Scan Line Image Strip Frame Buffer

Figure 2: The Influence of Cell Size on Vector to Raster Conversion Algorithms on the Synthetic Test Image. The width of each pixel is 20 units; the diameter of polygon A is 12.5 units. Note the filling errors in the Image Strip and Frame Buffer algorithms, and the abbreviation of the spines by the Scan Line technique.

REFERENCES

1. Ackland, B.D. and N.H. Weste, 1981. "The Edge Flag Algorithm - A Fill Method for Raster Scan Displays", *IEEE Transactions on Computers*. Vol. C-30, NO. 1, pp. 41-47.

2. Bonfatti, F. and P. Tiberio, 1983. "A Note on the Conversion of Polygonal to Cellular Maps", *Computers & Graphics*. Vol. 7, No. 3-4, pp. 355-360.

3. Carlbom, I. and J. Michener, 1983. "Quantitative Analysis of Vector Graphics System Performance", *ACM Transactions on Graphics*. Vol. 2, No. 1, pp. 57-88.

4. Franklin, W.R., 1979. "Evaluation of Algorithms to Display Vector Plots on Raster Devices", *Computer Graphics and Image Processing*, vol. 11, pp. 377-397.

5. Hanley, J.T., 1982. "The Graphic Cell Method: A New Look at Digitizing Geologic Maps" *Computers and Geosciences*, Vol. 8, No. 2, pp. 149-161.

6. Muller, J.C., 1977. "Map Gridding and Cartographic Errors / A Recurrent Argument", *The Canadian Cartographer*. Vol. 14, No. 2, pp. 152-167.

7. Nagy, G. and S.G. Wagle, 1979. "Approximation of Polygon Maps by Cellular Maps" *Communications of the ACM*. Vol. 22, No. 9, pp. 518-525.

8. Pavlidis, T., 1979. "Filling Algorithms for Raster Graphics", *Computer Graphics and Image Processing*. Vol. 10, pp. 126-141.

9. Peuquet, D.J., 1981a. "An Examination of Techniques for Reformatting Digital Cartographic Data / Part 1: The Raster-to Vector Process", *Cartographica*, Vol. 18, No. 1, pp. 34-48.

10. Peuquet, D.J., 1981b. "An Examination of Techniques for Reformatting Digital Cartographic Data / Part 2: The Vector-to-Raster Process", *Cartographica*, Vol. 18, No. 3, pp. 21-33.

11. Piwowar, J.M., 1987. "Conversion Between Vector and Raster Format Data for Geographic Information Systems Applications". Unpublished Masters Thesis, Department of Geography, University of Waterloo. 189 pp.

12. Selden, D., 1985. "An Approach to Evaluation and Benchmark Testing of Cartographic Data Production Systems", *Proceedings of the 7th International Symposium on Computer-Assisted Cartography*. pp. 499-509.

13. van Roessel, J.W. and E.A. Fosnight, 1985. "A Relational Approach to Vector Data Structure Conversion", *Proceedings of the 7th International Symposium on Computer-Assisted Cartography*. pp. 541-551.