ALGORITHMS FOR THE REDUCTION OF THE NUMBER OF POINTS REQUIRED TO REPRESENT A DIGITIZED LINE OR ITS CARICATURE

DAVID H DOUGLAS AND THOMAS K PEUCKER

University of Ottawa/Simon Fraser University, British Columbia

ABSTRACT. All digitizing methods, as a general rule, record lines with far more data than is necessary for accurate graphic reproduction or for computer analysis. Two algorithms to reduce the number of points required to represent the line and, if desired, produce caricatures, are presented and compared with the most promising methods so far suggested. Line reduction will form a major part of automated generalization.

Lines from maps and photographs are recorded numerically for cartographic manipulation to facilitate their reproduction at different scales and projections, and to allow map compilation with other geographic data bases. Usually lines are approximated by straight line segments, and end points of which are recorded by a pair of co-ordinates in either polar or orthogonal measure. The other more important methods by which lines are recorded are chain encoding and skeleton encoding. Chains approximate lines by a sequence of end to end vectors, where the length and direction of the vectors are selected from a fixed, usually four or eight, number of possibilities.[1] Skeleton encoding is directed more at recording closed areas or polygons by filling the area with circles or rhombi of different sizes. The lines forming the boundaries are recorded by implication.[2] The conversion of graphic data to computer readable numerical forms is effected with a co-ordinate digitizer, a bit plane scanner or an automatic line follower. A co-ordinate digitizer converts a pointer's location on a table to x-y values which can be written on punched cards or magnetic devices. Polar co-ordinate digitizers, which consist of a slide in a rotating anchor head, record a radius and an angle from a base vector. Another digitizing device consists of a pointer suspended from a pair of retracting wires which activate potentiometers. Conversion of values in one recording co-ordinate system to another can be performed easily with small computer programs.

Drum scanners superimpose a vast and very fine grid over the document to be digitized recording a "yes-no" or "on-off" value for each cell location, depending on whether that cell covers a line or not. A trade-off is introduced between the fineness of the mesh, implying more computer processing time to reduce the data to forms which are easily handled, and coarseness of the image recorded. On the other hand, the mesh density, being dependent on hardware, is fixed at the time of manufacture and is usually set to be somewhat smaller than the minimum line width. In all cases, the reduction of a bit plane scan, in which lines are represented by clouds of cells containing numerous discontinuities, to chain or vector encoded lines, is a complex process requiring processing time and resources which could only be described as being quite substantial.

With a co-ordinate digitizer lines may be recorded in point mode, time or increment automatic modes. Lines recorded in point mode are effectively generalized by the operator who subjectively selects points which best approximate the line to the degree he desires. This presumes, among other things, that he is his own customer. Point digitizing is extremely tedious however, and is unsuitable for anything but the simplest data sets, such as the generalized

---

David H. Douglas is a lecturer in Geography at the University of Ottawa and is presently on a study leave at Simon Fraser University, Burnaby, British Columbia: Thomas K. Peucker is an Associate Professor at Simon Fraser University, Burnaby, British Columbia.

MS submitted June 1973

[1]H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", *Institute of Radio Engineers; Transactions on Electronic Computers*, Vol. EC-10, 1961, pp. 26–268.

[2]J. R. Pfaltz and A. Rosenfeld, "Computer Representation of Planar Regions by Their Skeletons", *Communications of the ACM*, Vol. 10, No. 2, February 1967, pp. 119–122 and 125.

outlines of counties or census tracts. Most coordinate x-y digitizers on the market possess, as options, time or increment automatic recording modes. Points are recorded automatically in a given time interval, or after the cursor has moved a preset distance along the x and/or y axis. The prime limiting factor on the speed of recording is the speed of the output device. Magnetic tape transports which record up to 300 characters per second are commonly available, allowing up to 20 or 30 points to be recorded each second. To record coastlines, contour lines, or other lines of high frequency oscillation it is evident that the minimum speed required, given the speed at which an operator can follow a line, is in the order of 5 to 10 points per second, which effectively eliminates paper tape and punched cards as output media. Digitizing onto magnetic tape has more than its share of problems, primarily because there are no foolproof means to ensure the data are correctly recorded at the time of digitizing, and because of the inordinately frequent occurrence of non-confirmable digitizing errors, such as line ends which should, but do not meet, ... lines recorded twice and so forth. The editing procedures necessary are time consuming and clumsy. These problems have been met by elaborate on-line procedures where a mini-computer interfaced to the digitizing table oversees the whole operation, checks and double checks the data recorded, closes loops and signals when it senses a great many errors, such as cursor movement too fast to be accurate.[3]

All digitizing methods, except perhaps for the possible exclusion of point digitizing on a co-ordinate digitizer, record, as a general rule, far more points than necessary to reproduce the line on most graphic devices, even at the scale and resolution of the original line. The elimination of data representing unnecessary points, such as duplicates,

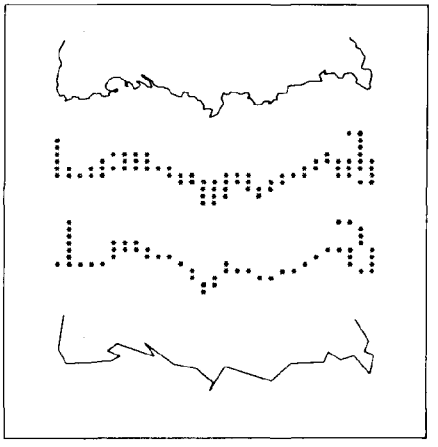

Figure 1. Line represented by 140 points on the plotter and the printer, and the same line represented by 25 points.

and points along a straight line, can be of significance, simply because of the diminished storage requirements. As well, the operating speed of many spatial analysis programs and the plotting speed of many graphic devices are related inversely to the number of points to be processed or plotted. Reduction of a line by elimination of unnecessary points representing it assumes a more positive advantage if the line is to be abstracted or caricatured purposely, if the scale of reproduction is to be smaller, or if the output device, such as some Cathode Ray Tube plotters, has a cruder resolution than represented by the original digitized line. Lines which have a higher frequency of oscillation than can be represented within the resolution capability of the graphic device become fuzzy and weak.[4] Figure 1 illustrates line data at the resolution of recording, its reproduction on the computer printer, and the reproduction of a greatly generalized version of the line. Given the crudeness of the printer as a graphic device it is evident that the simplified version of the line is preferable to the unsimplified one mainly because of the elimination of most

[3]A. R. Boyle: Computer Aided Compilation, Hydrographic Conference, Ottawa, January, 1970.

[4]and are similar in effect to the data clouds recorded by a bit plane scanner.

of the double lines and data clouds. Since this line was better represented by 25 points than it was by the original 140 obviously some computer pre-processing was justified.

There have been a great many approaches suggested and algorithms programmed to reduce the number of points required to represent numerically recorded lines. Some of these are in regular use within planning agencies and cartographic units. Not all of the methods have been exhaustively tested to measure or judge their cartographic usefulness and there have been few, if any, studies to compare the methods with each other. The methods can be classed broadly into the categories of: elimination of points along the line by one or more of a multitude of criteria; approximation of the line with a mathematical function; and deletion of specific cartographic features represented by the line. Of these categories, it would seem that the last one would come closest to duplicating the task as performed by an experienced cartographer as he generalizes.

The cartographer attempts to maintain the character and overall impression of an empirically defined, or hand drawn line by selective deletion of some of the details. A fjorded coast is represented by only a few of the actual number of fjords, a delta by only a few of the actual number of channels and so forth. The automation of this approach would rely therefore on the ability to program the computer to recognize specific cartographic features. One attempt is based on an interactive computer program which has the ability to "learn" from the actions of an operator.[5] The operator generalizes a line plotted on a cathode ray screen by signaling the deletion or maintenance of points. As the computer "learns" from what the operator selects it attempts to recognize similar features on its own. This system at its present level of development concentrates on the angular and length relationships of a very small number of segments, but the number of possible ways to represent a single simple class of feature, such as a peninsula, is simply staggering. This interactive system, therefore, represents but a small step towards the solution of a fantastically complex problem.

The second group seeks to approximate the points along a line with mathematical functions. This can be done for the whole line at once or it can be done in some piece-wise order taking a small number of connected points at a time. There are several different methods fitting into the latter category. One developed by A. R. Boyle for the Hydrographic Survey of Canada (1972) computes a first order least squares line through a fixed number of points and then steps forward in that direction by a predetermined distance. Two other approaches begin by defining the ends of segments as averages of a fixed number of points along the line. Koeman and Vander Weiden[6] suggest taking the mean while Jancaitis and Junkins[7] take the distance weighted centroid. When these central points are joined the results simulate a piece-wise approximation with functions of the first order. It must be mentioned, however, that the stated purpose of Jancaitus and Junkins was to smooth and not necessarily to reduce the line.

The resulting data sets of extracted functions are economical in terms of storage

[5]Andrew H. Clement, "The Application of Interactive Graphics and Pattern Recognition to the Reduction of Map Outlines", Master's Thesis, University of British Columbia, 1973.
[6]C. Koeman and F. L. Vander Weiden, "The Application of Computation and Automatic Drawing Instruments to Structural Generalization", *Cartographic Journal*, Vol. 7, No. 1, June 1970, pp. 47–49.
[7]James R. Jancaitis and John L. Junkins, *Mathematical Techniques for Cartography*, Final Contract Report for U.S. Army Engineers Topographic Laboratory, Fort Belvoir, Virginia, Contract No. DAAK02–72–C–0256, February 1973, pp. 15–20.

space required, but are relatively time consuming in the processing stage. The greater the number of points, the more costly and complex the operation. These functions reproduce lines which are typically much smoother than the lines they represent. In the main they are probably much better suited for smoothing than reduction and have to be considered of limited value for generalizing. Functions extracted in a piece-wise fashion tend to under-represent erratic curves and over-represent smoother curves. Methods which look for central tendencies are inclined to depress the effect of extreme points. Unfortunately, these are often the very points which give character to the line.

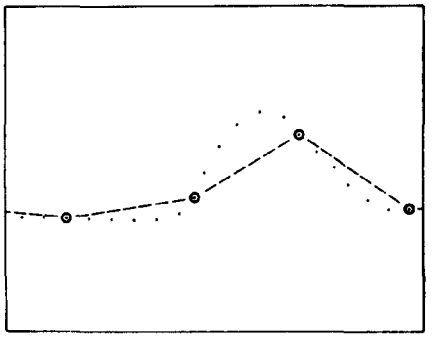

Figure 2. Line reduction by the selection of every sixth point.

Of the group of methods which eliminate points, some concentrate on the points which are to be deleted while others are directed towards selecting those points which are to be maintained. The algorithms directed at deleting points are usually the simplest. In the case of data recorded by time-automatic digitizing a simple test to drop those closer than one resolution unit can eliminate a large percentage of the points recorded. This method can be extended by purposely decreasing the numerical resolution or by establishing a threshold distance. Points closer than this distance to neighbours are dropped.[8] For chain encoding a simple compression on the basis of consecutively equal vectors can also result in significant savings. This can be extended as well for other types of encoding by dropping points whenever the direction of the line is not changed through a threshold angle by the segments subtended on it. The underlying purpose of these methods is to eliminate wasted data space but since the line plotted after this kind of processing would look very much the same as it would before it cannot represent a significant step towards automated generalization.

The simplest and most often used method of line reduction is to delete all but every n^{th} point along the line where n is a fixed integer based upon the desired degree of reduction.[9] The method does not require much in the way of computing resources and it furnishes acceptable results if the digitizing was extremely dense. The primary disadvantage is the frequent elimination or misrepresentation of important features along the line such as promontories, indentations, sharp angles and so forth. A secondary limitation is that straight lines are still over-represented. These shortcomings are made obvious in Figure 2.

The alternative to deleting points is to select them. In the special case of monotonically increasing lines (for instance, just one value of "y" for every "x"), crests and troughs may be selected. The obvious disadvantage here is the omission of points where there is a change of direction but which nonetheless are not crests or troughs. For irregular planar curves, the problem is more difficult. Jarvis con-

[8]W. R. Tobler, "Numerical Map Generalization", *Michigan Inter-University Community of Mathematical Geographers, Discussion* Paper No. 8, Department of Geography, University of Michigan, January 1966.

[9]Experimental Cartographic Unit, Royal College of Art: *Automatic Cartography and Planning*, London, Architectural Press, 1971.

verts the Cartesian to polar co-ordinates and then looks for crests and troughs.[10] This is useful for curves which can be made monotonic by this conversion, but, as for Cartesian measure, the solution cannot be considered general.

One alternative to line generalization which seemed to hold conceptual promise was that method provided by the German firm, A. E. G. which supplied the Experimental Cartographic Unit with its GEAGRAPH 4000 plotter, and was described by T. Lang in 1969.[11] This method was reported as producing acceptable results but was eventually rejected as a general purpose technique by the Experimental Cartographic Unit on the grounds that it required far too much computer time for the on-line processing system being operated at the time. The objective of the procedure was to delete points if they were found to lie within a tolerance distance of a straight line segment being tested to represent a portion of the line. From one representative point it constructs straight lines to subsequent points until one point between the representative point and the sub-point is further away from the line linking the two than a pre-set tolerance value. As soon as this condition is satisfied, the point before the sub-point becomes a new representative point and the procedure is repeated. The method gives acceptable results in the case of smooth curves but it does not detect the best representative points on sharp curves and the results are particularly unsatisfying where sharp angles are numerous.

The methods proposed in this paper are based on a concept somewhat similar to the pre-set tolerance ideas described by Lang but concentrates rather on the selection of points rather than on their deletion.

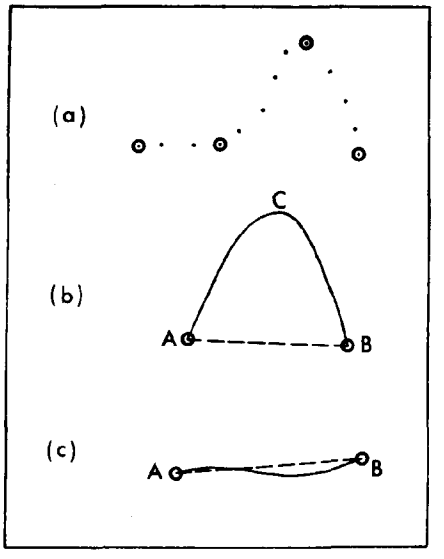

Figure 3. Subjective selection of representative points.

Approaches to a computerized solution to many problems begin with an examination of the way one would solve them subjectively. Consider the line represented by points illustrated in Figure 3(a). One might choose the encircled points as those which represent the original line to our own requirements of accuracy. Perhaps the reason we would select these points and not others might be illuminated by examining the simpler situations in Figure 3(b) and (c). Starting with the obligation to begin with the end points, the question might be: "Why would there be a compulsion to insert a point C in (b), where no such compulsion would exist in (c)?" The perpendicular distance of C from the segment A-B may provide a clue. This suggests that an arbitrary maximum distance could be established. If no point along the line is further than this distance from the straight line segment connecting its end points, then the straight line seg-

[10]C. L. Jarvis, "A Method for Fitting Polygons to Figure Boundary Data", *The Australian Computer Journal*, Vol. 3, 1971, pp. 50–54.
[11]T. Lang, "Rules for Robot Draughtsmen", *Geographical Magazine*, Vol. XLII, No. 1, Oct. 1969, pp. 50–51.

ment will suffice to represent the original line. If this condition is not satisfied, then another point along the curved line must be selected and the same test would be carried out with the new segments. The next question is: "What point along the curved line should be selected to become the end point of the two new straight segments created?" The obvious answer is the furthest point from the straight segment. Although it is possible that this point may be embedded in a long smooth curve, it is more likely that it is the apex of a relatively sharp angle. As well, this point has already been identified as a result of the distance search, therefore, the benefits associated with its selection far outweigh the possible attraction of selecting some other representative point. In the case of closed loops, where the first and the last point do not define a line then the maximum perpendicular distance from the segment is replaced with the maximum distance from the point. The same process would be repeated with the new segments created until the maximum distance requirement is satisfied for all straight segments.

Two different procedures embodying these principles have been encoded in FORTRAN IV and tested. In addition Method 2 has been encoded as a recursive function in ALGOL W.[12]

Method one begins by defining the first point on the line as an anchor and the last as a floating point. These two points define a straight segment. The intervening points along the curved line are examined to find the one with the greatest perpendicular distance between it and the straight line defined by the anchor and the floater. If this distance is less than the maximum tolerance distance, then the straight segment is deemed suitable to represent the whole line. In the case where the condition is not met, the point lying furthest away becomes the new floating point. As the cycle is repeated the floating point advances toward the anchor. When the maximum distance requirement *is* met the anchor is moved to the floater and the last point on the line is reassigned as the new floating point. The repeat of this latter operation comprises the outer cycle of the process. The points which had been assigned as anchor points comprise the generalized line.

Method two is exactly the same as method one except that note is taken of all points which have been assigned as floaters on previous inner cycles. These are stacked in a vector. After the anchor point is moved to the floating point, the new floating point is selected from the top of this stack, thereby avoiding the necessity of re-examining all the points between the floater and the end of the line. This procedure usually results in the selection of a slightly greater number of points than Method 1, but takes approximately 5 per cent of the computing time and is thought to produce better caricatures. This method can also be thought of as taking a logically hierarchical approach to line reduction. On one cycle extreme points are selected and these tested to see if they suffice. If they do not, intermediate points are taken and the same question asked about each of the two new segments produced, and then each of the four new segments are examined, ... and so on as if in a branching tree. Each branch is terminated when the offset tolerance criterion is satisfied.

To enable valid comparisons four separate subroutines were written on the basis of the procedure described by Lang. One was an exact duplication of that procedure while the other three were combinations of two incorporated modifications.

---

[12]Andrew H. Clement, "The Application of Interactive Graphics and Pattern Recognition to the Reduction of Map Outlines", Master's Thesis, University of British Columbia, 1973.

The program Lang describes starts by assigning the first point as the anchor and third as a floater. The second is tested to see if it lies within tolerance distance of the segment defined by the anchor and the floater. If it does, the fourth is assigned as the floater and the second and third are examined and so on. The first floating point defining a segment which does not allow all intervening points to satisfy the tolerance criteria causes the anchor to move to the point before the floating point. Since selection of the point immediately before the floating point has no cartographic justification, the first modification of the procedure has the anchor point move to the point furthest from the segment. The reasoning behind selecting the furthest point is that it is the one most likely to subtend a sharp angle and would therefore have the best chance of properly representing the line.

The second modification attempts to cut computing time by avoiding unnecessary repeated calculations of distance. From Figure 4, it is clear that in most cases, the sum of the distances a + b + c is greater than the greatest distance that P_1, P_2, or P_3 lies from the segment P_0P_4. In other words, if a + b + c is less than the tolerance distance then d also would be less than the tolerance. Only one distance, rather than all of the intervening ones, has to be calculated on each cycle. The inner cycle, intended to find the point lying furthest from the segment, is invoked only in the cases where the accumulated total is greater than the tolerance. Positive and negative accumulations are kept in separate registers to avoid subtractions from their absolute magnitudes in the case of double curves. The maintenance of these running registers is particularly useful when series of points lie along straight lines.

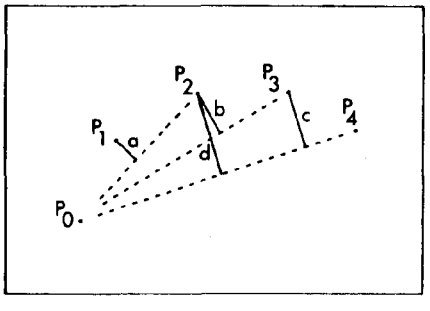

Figure 4. Running registers of accumulated offset distances.

The first modification which attempts to select a point which is more rationally defined than simple convenience, has the expected result of approximately doubling the number of points selected and the processing time required to isolate them. The second modification definitely reduces the time required to process a given line, especially if a great many points are deleted because they lie along relatively straight segments.

All procedures were tested and compared, both for their ability to remove unnecessary points, that is with the offset tolerance set to be less than the resolution of the plotting device (Figure 5), and their ability to produce caricatured representations (Figure 6). All were judged to produce satisfactory results for simple line reduction, however the versions of the A. E. G. procedure without the modification to pick the furthest point from the tested segment did not produce satisfactory caricatures because of the tendency to omit and cut corners. The methods presented in this paper were tested with substantial data sets and found to be operationally suitable both for simple reduction and in the production of satisfactory abstractions (Figures 5 and 6).

Detailed comparisons in computing time required for each sub-routine were made on the basis of a three inch square and a three inch diameter circle, each made up of 4000 points evenly distributed along its periphery. It was felt that the square would give ample opportunity to demonstrate the power of each routine in the

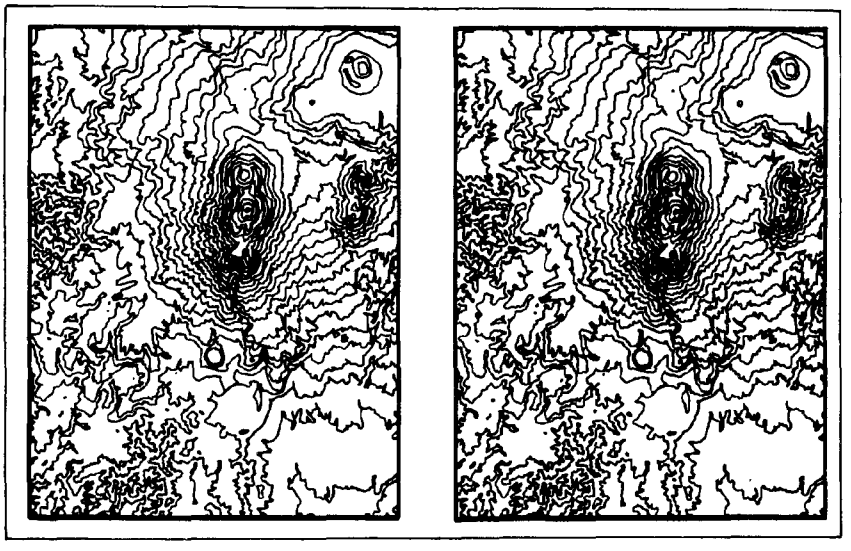

Figure 5. Contours plotted from the original digitized, unduplicated at .001 resolution, 41,311 points (left), and from 7,782 points (right) reduced by Method 2 with a tolerance set to half the resolution of the plotter. The reduction procedure added 16.5 seconds to the 64 seconds required to read and write the data to plot the map on the left. The images may be compared with a simple stereoscope.

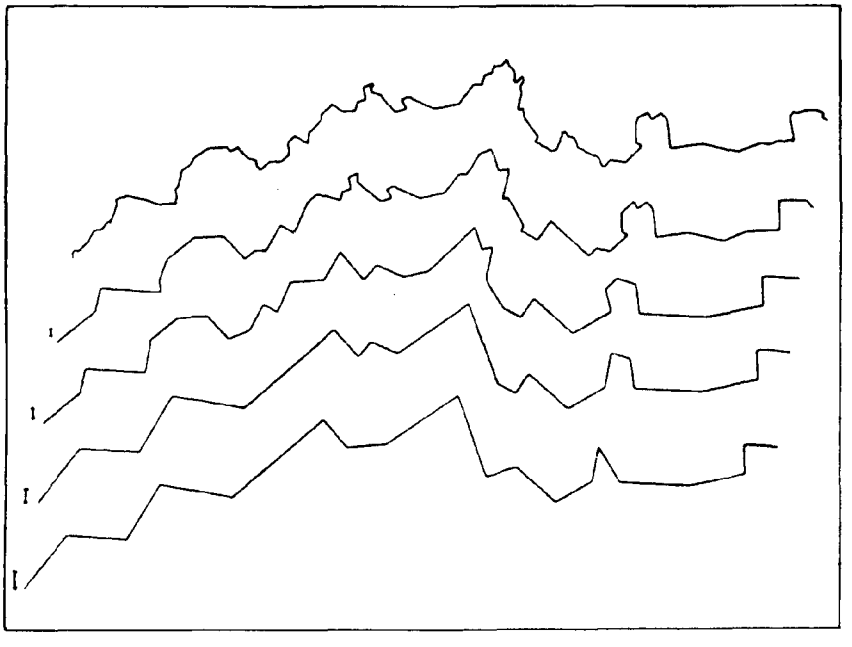

Figure 6. Line reduced and caricatured by Method 2. The tolerance value employed is shown to scale at the left of each caricature which was reduced from the original data represented by the top line.

TABLE 1

PROCESSING TIME REQUIRED TO REDUCE A 3″ CIRCLE AND A 3″ SQUARE, MADE UP OF 4000 POINTS EACH EVENLY SPACED ALONG THE PERIMETER, TO THE NUMBER OF POINTS INDICATED WITH THE GIVEN OFFSET TOLERANCE.

	Offset Tolerance (inches)											
SQUARE	.001		.005		.01		.05		.1		.5	
4000 points	Points	Time	Points	Time	Points	Time	Points	Time	Points	Time	Points	Time
A.E.G. procedure	5	88.4	5	88.6	5	87.3	5	88.3	5	86.4	5	86.9
A.E.G. plus Mod. 1	5	87.8	5	88.9	5	88.6	5	88.3	5	89.8	5	113.9
A.E.G. plus Mod. 2	5	22.6	5	44.5	5	46.0	5	45.8	5	44.5	5	44.7
A.E.G. plus Mods. 1 and 2	5	22.8	5	22.5	5	22.4	5	22.9	5	23.1	5	31.4
Method 1	5	.7	5	.7	5	.8	5	.8	5	.8	5	.7
Method 2	5	.6	5	.6	5	.6	5	.5	5	.5	5	.6
CIRCLE												
4000 points												
A.E.G. procedure	88	5.5	40	11.1	29	14.9	14	32.6	10	42.4	5	97.1
A.E.G. plus Mod. 1	171	10.4	77	20.4	55	28.6	25	60.4	18	84.8	5	109.4
A.E.G. plus Mod. 2	88	5.4	40	10.8	29	15.4	14	32.7	10	41.6	5	92.2
A.E.G. plus Mods. 1 and 2	171	10.6	77	21.5	55	30.0	25	60.9	18	87.8	8	170.2
Method 1	127	25.1	56	10.4	39	7.5	18	3.7	13	2.7	6	1.0
Method 2	129	1.8	65	1.5	33	1.2	17	.9	17	1.0	5	.6

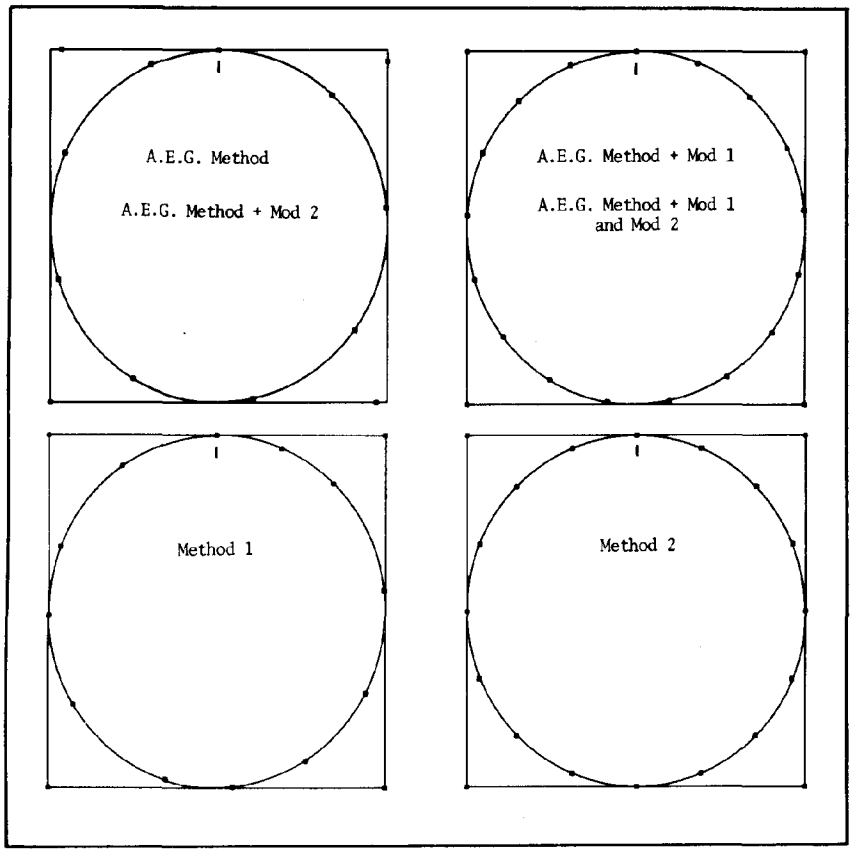

Figure 7. Plotted results for a circle and a square each made up of 4,000 points around its perimeter. The dots on the boundaries indicate the selected points by the indicated procedures. The tolerance value of .1 inch is illustrated to scale at the top centre of each diagram.

case where many points along a line are to be deleted, whereas the circle would be more representative in sinuosity of drawn or empirically recorded lines as far as this timing test was concerned. Table 1 presents results by the established offset tolerance in number of points selected and in seconds central processing unit time required for the reduction procedure (I. B. M. 370/155 under O/S MVT).

Figure 7 illustrates plotted results for a 1/10″ offset tolerance with the points selected by each routine marked with a heavy dot. The fact that the routines selected different points and different numbers of points is not unusual or unexpected and similar differences occur in the case of sinuous lines. The shortcoming of the unmodified A. E. G. procedure is evident in the case of points selected to represent the square, ... which were just less than one tolerance unit from the corners for all but the first and last point.

Each routine selected five points to represent the square and each took approximately the same time regardless of the tolerance, except with the second modification. In this case the first step off the straight line caused the inner cycle to be invoked which found the new anchor

point on the first iteration. More iterations were required for the other tolerance limits.

In the case of the circle an increase in tolerance limit caused a decrease in the number of points found to represent it for all methods. Those methods which push the examination segment ahead of the anchor points, that is the A. E. G. method with none, one or two modifications, take longer to perform as the offset tolerance is increased. This therefore comprises the main reason that they have to be considered unsuitable in an operational context. These procedures are fastest if they are unable to delete any points, because in such cases they would have to examine only one point to come to that decision. On the other hand if a great number of points are found to be deletable, increasingly large inner cycles are invoked for each advance of the floating point. The two methods presented in this paper work in entirely the opposite way and are fastest in the case of lines which are found to be representable with a smaller number of points. Presumably this is the object of the effort. In all cases Method 2 is seen to take as little as 1 per cent of the time required by the others.

The prime purpose of the routines discussed here is to reduce the number of points required to represent a line and to produce abstractions, or caricatures of the line in cases where these will suffice. In many cases these could be considered to be perfectly adequate generalization procedures. While the scope of generalization is no doubt much broader, line reduction by means such as those described here, represents an important portion of that topic.

RÉSUMÉ. Règle générale, les méthodes numériques enregistrent des lignes avec beaucoup plus de données qu'il n'est nécessaire à la reproduction graphique précise ou à la recherche par ordinateur. L'auteur présente deux algorithmes pour réduire le nombre de points nécessaires pour représenter la ligne et produire des caricatures si désiré, et les compare aux méthodes les plus prometteuses suggérées jusqu'ici. La réduction de la ligne constituera une partie importante de la généralisation automatique.

ZUSAMMENFASSUNG. Alle Digitalisierungsmethoden zeichnen in der Regel Linien mit bedeutend mehr Daten auf als für eine genaue graphische Wiedergabe oder für eine Computeranalyse notwendig sind. Zwei spezielle Rechenverfahren zur Reduzierung der Punktezahl, die zur Darstellung einer Linie benötigt werden und die auch falls erwünscht Verzerrungen produzieren, werden vorgstellt und verglichen mit den bisher am meisten versprechenden Methoden. Die Linienreduzierung wird eine grosse Rolle in der automatisierten Generalisierung spielen.

RESUMEN. Todos los métodos digitales, como regla general, registran líneas que tienen mucho más datos que los necesarios para la reproducción gráfica correcta o para el análisis por computadora. Se presentan dos algorítmos para reducir el número de puntos necesarios para representar una línea y si se desea, producir caricaturas; estos se comparan con los métodos más pometedores sugeridos hasta ahora. La reducción de líneas formará gran parte de la automatización en general.