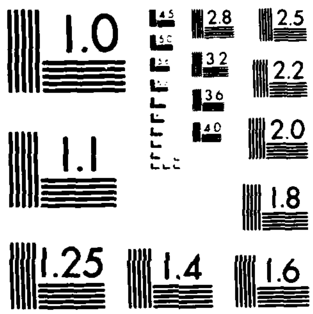
AD-A143 250 MANUAL AND AUTOMATED LINE GENERALIZATION AND FEATURE DISPLACEMENT(U) ZYCOR INC AUSTIN TX S ZORASTER ET AL. JUN 84 101-43-16 ETL-0359 DAAK70-82-C-0149 1/2

UNCLASSIFIED F/G 5/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

ETL-0359

MANUAL AND AUTOMATED LINE GENERALIZATION AND FEATURE DISPLACEMENT

Steven Zoraster
Dale Davis
Marc Hugus

ZYCOR, Inc.
2101 South IH-35
Austin, Texas 78741

JUNE 1984

Prepared for
U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060

ETL

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: ETL-0359 — 2. GOVT ACCESSION NO.: AD-A143250	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): MANUAL AND AUTOMATED LINE GENERALIZATION AND FEATURE DISPLACEMENT	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report August 1982 — June 1984
	6. PERFORMING ORG. REPORT NUMBER: 101-43-16
7. AUTHOR(s): Steven Zoraster, Dale Davis, Marc Hugus	8. CONTRACT OR GRANT NUMBER(s): DAAK70-82-C-0149
9. PERFORMING ORGANIZATION NAME AND ADDRESS: ZYCOR, Inc. 2101 South IH-35 Austin, Texas 78741	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060	12. REPORT DATE: June 1984
	13. NUMBER OF PAGES: 121
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

Appendixes for this report are contained in ETL-0359-1.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automated Cartography	Line Smoothing
Feature Displacement	Map Compilation
Line Generalization	Map Generalization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The primary objective of this effort was to examine map compilation techniques used for manual and automated line generalization and feature displacement. Manual techniques used at several Defense Mapping Agency compilation sites are described in detail. An extensive survey of algorithms suggested in the cartographic, image processing, and computer science literature is provided. Possible criteria for evaluating algorithm performance are discussed and line generalization algorithms are ranked according to certain of these measures.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

This document was prepared under Contract DAAK70-82-C-0149 for the U.S. Army Engineer Topographic Laboratories, Fort Belvoir, Virginia 22060 by ZYCOR, Inc. The Contracting Officer's Technical Representative was Mr. Douglas Caldwell.

Report No. 101-43-16

MANUAL AND AUTOMATED LINE GENERALIZATION AND FEATURE DISPLACEMENT

Final Report

Steven Zoraster
Dale Davis
Marc Hugus

Submitted to:

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORY
Fort Belvoir, VA 22060

Contract No. DAAK70-82-C-0149

Submitted by:

ZYCOR, INC.
2101 South IH-35
Austin, Texas 78741

June 30, 1984

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

1.0 INTRODUCTION

This final report is provided to the U.S. Army Engineer Topographic Laboratories (ETL) to complete work required under contract DAAK70-82-C-0149. In that contract, ZYCOR was directed by ETL to study line generalization and feature displacement for the Defense Mapping Agency (DMA). Specifically, ZYCOR was directed to perform the following tasks:

- analyze DMA manual and automated techniques, specifications, and requirements,
- analyze non-DMA algorithms and software for line generalization and feature displacement, and
- make recommendations for automating line generalization and feature displacement.

In February 1983 an interim report was produced to satisfy the first contract task. It described current techniques of line generalization and feature displacement used by DMA cartographers. Most of the information used in that report was obtained during visits to DMAAC, DMAHTC, and the DMAHTC Field Office in San Antonio. Other material was obtained by reviewing DMA manuals, internal reports, map specification guides and contractor compilation guides.

In September 1983 a second interim report was produced to satisfy the second contract task. It described techniques for line generalization and feature displacement used or suggested by non-DMA sources. Information used in that report was obtained by a review of articles published in cartographic, geographic, and computer science journals as well as contacts with university and research cartographers.

This final report combines the significant material produced in the first two reports and contains contract conclusions along with recommendations for further work in automated line generalization and feature displacement.

1.1 DMA AUTOMATION REQUIREMENTS

Currently many small scale DMA maps and charts are created by manually generalizing large scale sources. This is a slow, manpower intensive process which cannot meet the throughput requirements expected in the 1990's. Furthermore, in the future, DMA intends to produce maps from digital cartographic data bases compiled at one or more base scales. These data bases will be accessed to produce maps at any scale using digital techniques.

Automation of cartographic processes is required if DMA is to meet the expected demand for its products and fully utilize its data base capabilities. In support of this effort, algorithms for line generalization and feature displacement are required.

Automation of line generalization and feature displacement is desirable for reasons beyond the need to increase map production or the desire to create a fully automated production process. Generalization, done manually, is a very subjective task. This may result in products which, although satisfying strict DMA specifications, are internally inconsistent due to variations in personal interpretations and skill. The use of a fixed set of generalization and displacement algorithms will assist in the production of maps with uniform information content.

In a manual environment, an additional problem may arise when previously generalized maps are used as the source for smaller scale maps. This presents the opportunity for errors to propagate from both original source and intermediate maps to the final product. This is an undesirable, yet inherent problem in manual cartography. It is currently regulated by time consuming quality control procedures. By eliminating intermediate compilation steps and the associated cartographic license, automated techniques may be able to improve overall map accuracy.

Finally, current map production at DMA is a specialized process with unique experience and training required to produce each product. This restriction on the assignment of cartographers limits the ability of DMA to produce a variety of maps at a high throughput rate in crisis conditions. With algorithms to perform standard tasks while meeting varied map specification guidelines, DMA will have more flexibility in the use of its human resources.

1.2 SUMMARY OF RESULTS

This contract provides DMA with the following:

- a written description of generalization procedures currently used in manual map compilation at DMA
- an up-to-date survey of techniques for automated line generalization and feature displacement
- evaluations and recommendations of likely algorithms for automation
- recommendations for further work in automated line generalization and feature displacement.

1.3 ORGANIZATION OF REPORT

Chapter 2, "The Line Generalization and Feature Displacement Problem", provides an overview of line generalization and feature displacement cartography along with a discussion on the various definitions of these task used by DMA and other cartographers.

Chapter 3, "Map Generalization at DMA", discusses techniques used to guide line generalization and feature displacement at various DMA facilities.

Chapters 4 and 5, "Line Generalization Algorithms" and "Feature Displacement Algorithms", discuss algorithms which may be used for the generalization of linear cartographic data and for the displacement of cartographic features. This material was gathered on the basis of an exhaustive review of cartographic and computer science literature.

Section 6, "Evaluation of Generalization Algorithms", provides a set of cartographic and computational measures for evaluating algorithm performance. Certain of the evaluation criteria are used to judge and compare algorithms in Chapter 7, "Conclusions".

Ten appendices contain supporting material. Appendix A, "Bibliography", lists all reference sources used in performing this study. These references are repeated in Appendix B, "Keyed Bibliography", where they are organized into sections that match the algorithm ordering in Chapters 4 and 5.

Appendix C, "Compilation Guidelines", describes technical documents used at DMA to specify standards for map production. Methods by which these standards are applied for certain specific products are described in Appendix D, "DMA Compilation Procedures".

DMA has a number of computer systems which are used in various stages of the map compilation process. Some of these are described in Appendix E, "DMA Automated Technology".

As part of this study a survey on line generalization and feature displacement was presented to DMA cartographers. The survey along with an analysis of the results provided in Appendix F, "Map Generalization Survey". The problem of selecting features for map display was often encountered when discussing map generalization; this subject is discussed in Appendix G, "Selection".

ZYCOR contacted many non-DMA cartographers to become familiar with research being performed in map generalization outside of the government. The information we obtained is summarized in Appendix H, "Current Research and Development by Non-DMA Cartographers".

Recommendations for further research in line generalization and feature displacement are provided in Appendix I, "Recommendations for Future Research".

Finally, definitions for the technical terms used in this report are provided in Appendix J, "Glossary of Terms Relating to Line Generalization, Feature Displacement, and Cartography".

2.0 THE LINE GENERALIZATION AND FEATURE DISPLACEMENT PROBLEM

This section discusses map generalization and the definitions of line generalization and feature displacement under which ZYCOR has worked.

2.1 REASONS FOR GENERALIZATION

In compiling any map, decisions must be made as to what information is to be shown and how it is to be represented. When this process is used for the creation of small scale maps from larger scale cartographic sources in a way which requires reduction in detail, it also involves cartographic generalization.

The need for generalization comes from two different sources: 1) reduction of natural and cultural features in accordance with the scale of the map and the difficulty of symbolizing many of these features precisely at small scales; and 2) communication of the relationships underlying observations of geographic phenomena through elimination of unnecessary detail or exaggeration of important detail. (Taketa, 1978)

During scale reduction, generalization usually is thought of as smoothing character but this does not necessarily imply a reduction in the information content in a final product (Robinson, et al, 1978). In fact, generalization is a necessary component in maintaining a high level of general information on a map. Attempting to reduce scale without generalization can produce a map with a level of detail too great to be understood by a user.

Unfortunately, understanding the need for generalization does not lead directly to well defined procedures or even to commonly agreed definitions.

2.2 DEFINITIONS

The definition of map generalization used by the Department of Defense (DOD) differs considerably from those used widely in the cartographic community. *Department of Defense and* other definitions are discussed in this section.

2.2.1 Definitions Used by DMA

Map generalization is defined in the *Glossary of Mapping, Charting, and Geodetic Terms* (GMCG) as, "Smoothing of the character of features without destroying their visible shape. Generalization increases as map scale decreases." In this same source, *character* is *defined* as "the distinctive trait, quality, property, or behavior of manmade or natural features as portrayed by a cartographer. The more character applied to detail, the more closely it will resemble these features as they appear on the surface of the earth." A definition for line generalization may be obtained easily from the above generalization definition by replacing "feature" with "linear feature". Linear features include contours, drains, boundaries, and many cultural features.

Displacement is defined by the GMCG as "the horizontal shift of the plotted position of a topographic feature from its true position, caused by required adherence to prescribed line weights and symbol sizes." Displacement is an unavoidable result of the varied symbolization requirements and density of detail required for different map products.

The process of selecting features for display at a particular scale is regarded by DMA as a completely separate task. Appendix G describes research in the selection problem.

2.2.2 Alternate Definitions

Although the definition for feature displacement seems to be agreed upon within the cartographic community, there are many definitions for line generalization besides that provided in the GMCG (Steward, 1974). This variation among line generalization definitions can be attributed to at least three causes, 1) the richness of the English language, 2) the lack of understanding of the thought processes and evaluations associated with manual line generalization and 3) the variety of ways in which the task of line generalization is considered.

In the academic community, cartographers often refer to "generalization" as a broad collection of one or more of the following processes: simplification, classification, symbolization, and induction (Robinson et al., 1978). These are defined below:

> Simplification: The determination of the important characteristics of the data, the retention and possible exaggeration of these important characteristics and the elimination of unwanted detail.
>
> Classification: The ordering or scaling and grouping of data.
>
> Symbolization: The graphic coding of the scaled and/or grouped essential characteristics, comparative significances, and relative positions.
>
> Induction: The application in cartography of the logical process of inference.

3.0 DMA MANUAL MAP COMPILATION TECHNIQUES

This chapter contains a discussion of the line generalization and feature displacement procedures used during compilation of certain selected maps and charts at DMA. A firm understanding of the difficult manual techniques used in line generalization and feature displacement is required in order to evaluate a potential for automation. Visits to DMAAC, DMAHTC and the DMA field office at San Antonio provided the opportunity to observe skilled cartographers performing various generalization processes.

Four different map compilation sites were visited at the three facilities. Compilation procedures for Series 200 Charts, JOGS, and various nautical charts were performed at these sites. As discussed in Section 3.2 the general methods utilized at each site were similar. However, at each site ZYCOR had an opportunity not only to watch the compilation process, but also to engage in discussions with the cartographers involved. These sessions led to an interesting collection of comments on the manual compilation processes. These comments provide the most valuable part of this chapter. In certain cases the comments were supplemented by information obtained through a written survey provided to DMA personnel. The survey is described in detail in Appendix F. Appendix D provides details of compilation practices which are summarized in this chapter.

3.1 DMA OFFICES AND PRODUCTS

The Defense Mapping Agency produces maps, charts and digital information for use by the Armed Forces and all national security operations. DMA also produces nautical and aeronautical charts for a variety of non-military navigation purposes.

The two main offices of DMA provide mapping services directed toward different parts of the military. The Aerospace Center concentrates on aeronautical charts and digital information for aviation purposes. The Hydrographic/Topographic Center produces products primarily for tactical use by the Army and Air Force and navigation charts for use by the Navy and by non-military mariners. Both centers also are concerned with primary data collection from various sources. Field offices at a number of sites in the United States perform these same tasks under the general direction of one of the primary centers.

The maps and charts produced at DMA are created at many standard scales, including 1:50,000, 1:100,000, 1:200,000, 1:250,000, 1:500,000, 1:1,000,000, 1:2,000,000, and 1:5,000,000. Hydrographic charts are produced at a large number of scales which depend on the particular area being mapped.

3.2 GENERIC MAP COMPILATION PROCEDURES AT DMA FACILITIES

The DMA facilities are tasked with the generation of a large number of products. Final users, sources, required scales, cartographic projections, symbology and purpose vary widely over time between facilities and sections. Thus, encountering different map compilation procedures is to be expected.

On the other hand, the cartographic work observed by ZYCOR always involved the creation of small scale maps from large scale cartographic sources. Given that the inputs were primarily graphic and the final product was also graphic, it is not suprising that the general techniques utilized throughout were similar. Most differences fall into categories such as the ordering of operations, types and uses of supplementary information, emphasis on symbology and arrangements for quality control. This section is intended to provide the reader with a general overview of "standard" or "generic" map compilation and generalization procedures used when working from cartographic sources.

3.2.1 Map Preparation Guides

Map compilation for any product at DMA is initiated by the creation of a Map Preparation Guideline by the Scientific Data Division. A guide is prepared for each set of products. It contains unique requirements for the product, guidelines for the required horizontal control, cartographic projection, and a list of sources to be used, including cartographic, photographic and intelligence with priorities for their use and references to specific map specification guidelines for the scale of map desired.

3.2.2 Specification Guides

For a number of output products, DMA has final product specification guidelines which include detailed information on symbology, accuracy, selection of features, minimum feature separation, and annotation. These are designed to answer most of the common problems faced in compiling a map at a particular scale.

3.2.3 Sources

Source material is developed from a variety of data including graphic, photographic and textual materials. Examples of graphic data are maps, charts, plans and diagrams. Cartographic sources often include maps at 1:24,000, 1:50,000, and 1:62,500 from DMA and USGS sources and obtainable commercial or foreign government maps covering the same regions. Examples of these last sources include road maps produced by county governments, petroleum industry oil exploration maps, and maps created and supplied by national mapping agencies of foreign governments with which DMA has mapping agreements.

Photographic materials consist of stereo and monoscopic aerial photographic sources. Rectified photographs are desirable for their planimetric accuracy while unrectified photographs are useful for their high resolution.

Textual data include geodetic control memoranda, reports, population statistics, transportation time tables, geographic and geologic publications, periodicals and newspapers. Also within this category are a number of special intelligence sources.

3.2.4 Compilation

Compilation essentially begins with the creation of pull-ups. A pull-up is a graphic enhancement of selected cartographic features from source materials on a transparent medium which is laid on top of the source material. Using pull-ups, source detail is generalized in its relative position, then reduced photographically to the desired publication scale. The usual material for pull-ups is transparent mylar. Drawing is performed with a variety of pens, pencils, and felt tip markers. From each input map multiple pull-ups are created, including (depending on product) relief, drainage, cultural and vegetation overlays.

Figures 3.1 and 3.2, provided by DMA, show parts of a topographic source map with the original features overdrawn with highlighted lines as they would appear when drawn on a pull-up overlaying the source material. Much of the original data in these two figures has been significantly generalized. For example, in the upper left corner of Figure 3.1 rather large lakes have been eliminated from the drainage system. In Figure 3.2, streams have been included on the pull-up at the top of the map but eliminated at the bottom.

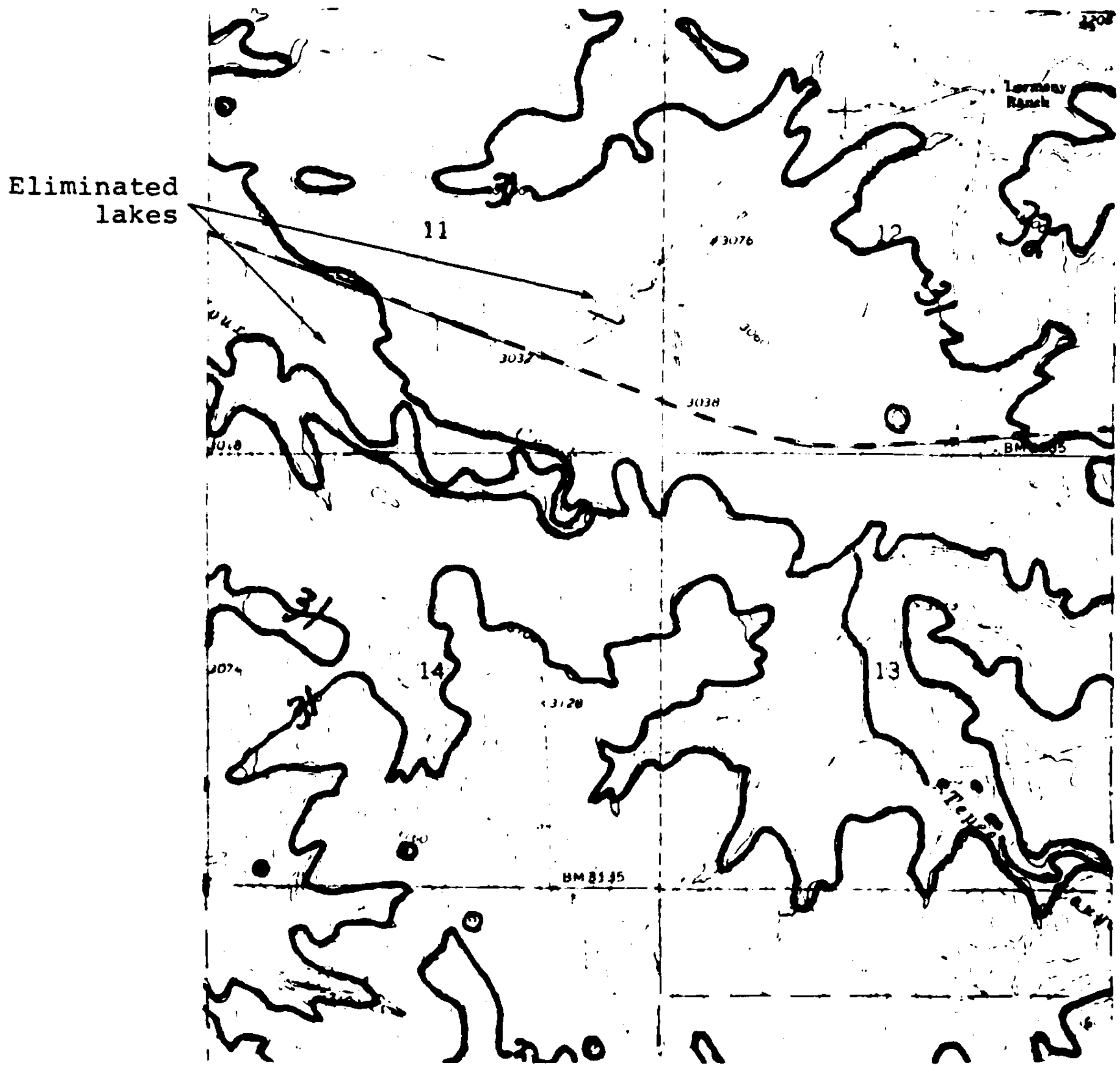


Figure 3.1. Sample Pull-Up

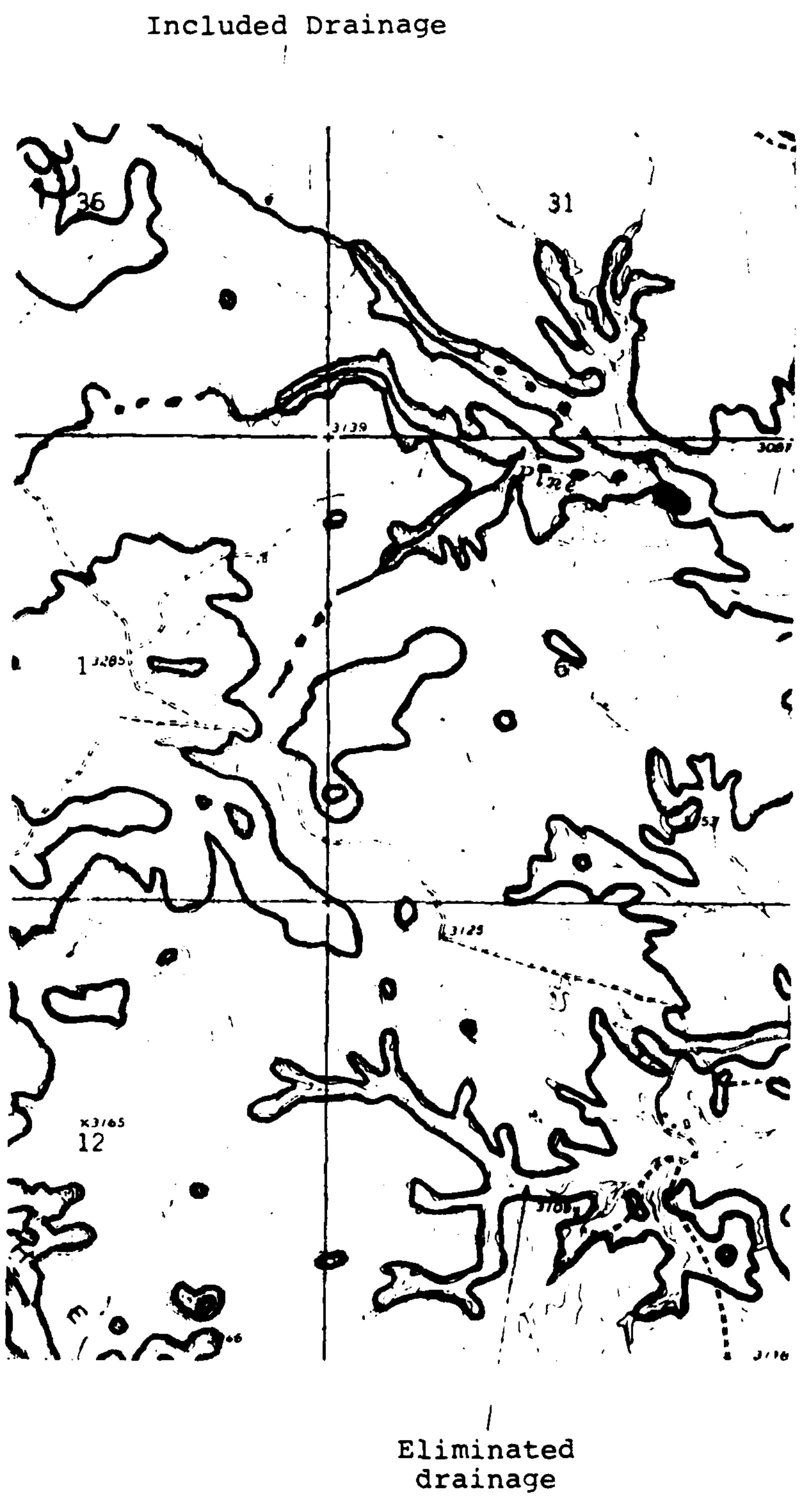


Figure 3.2. Sample Pull-Up

Most manual line generalization and feature displacement takes place at the pull-up creation phase. The difference between the scale of the source documents and the target scale along with the symbology requirements of the final product controls how much generalization and displacement is to occur in order to accurately and clearly represent the important features at the target scale. Templates and choice of line weights for drafting instruments can aid the compiler in performing this task.

Reference to map specification guides and previous products, along with advice from other cartographers, can provide useful guidelines in compilation and generalization. Ultimately, the amount of line generalization and feature displacement performed on a particular pull-up depends on the training and experience of the individual compiler. Inevitably two compilers working from the same sources according to the same specifications will produce slightly different final products.

Once a compilation step is completed, the pull-ups for each category are photographically reduced to the target scale and abutted together. At DMAAC, this process is referred to as panelling while at DMAHTC it is referred to as mosaicking. The photographic reduction produces the required change of scale. The placement of the reduced pull-up is guided by a geographic grid in the correct cartographic projection developed for each map.

The total number of pull-ups going into these mosaics can be quite large. If all sources are 1:24,000 and the final product is of roughly the same physical size at 1:250,000, each individual mosaic will contain 100 reduced pull-ups. Since there may be as many as four mosaics produced for each map, it may be necessary to create 400 pull-ups for a single product.

For some DMA products there may be a number of stages in which pull-ups are created, mosaicked, and photographically reduced. For example in the creation of Series 200 charts at DMAAC a two stage process was used. The first stage involved pull-up creation to be used in the reduction from source scale to 1:125,000; the second step, compilation from 1:125,000 to the final 1:200,000 scale. Two stages are also used in the creation of 1:300,000 scale nautical charts at DMAHTC when the original source material is 1:15000.

3.2.5 Engraving

The final mosaicked document is now used in an engraving process. From the mosaicked document, a film negative is produced for reproduction on to a scribecoat. The scribecoat is a thin opaque coating on a stable base material. The detail reproduced on the scribecoat is engraved manually and may be used to create the necessary peelcoats or open window negatives. These peelcoats are used for tinted areas of the map.

During the production of the scribecoat and peelcoats, a lettering sheet and negative for the map are also produced. Finally, composite negatives are created for each series of colors and a color proof is made to verify the registration and accuracy of the map. If no errors are found, the composite negatives are used to make printing plates from which the maps are lithographed.

3.2.6 Quality Control

Quality control may take place at a number of different stages. Of these, the easiest is at the original pull-up compilation level. Problems detected then can be easily corrected by the compiler. Needed line generalization and displacement which escaped the notice of the compiler may become clear at the engraving stage where the symbology is finalized. The amount of freedom the engraver has to handle problems varies widely from site to site. At some locations the engraver must call the attention of the compiler to all problems found. At others the engraver is expected to make minor changes himself.

3.3 SUMMARY OF COMMENTS PROVIDED BY DMA CARTOGRAPHERS

3.3.1 Line Generalization

It is extremely difficult to quantify the line generalization process or to measure the results of generalization. ZYCOR found that DMA personnel at one site felt that correct generalization was a natural result of adapting to the line weights required for correct symbolization on pull-ups while personnel at other sites relied on "cartographic judgement" e.g., on intuitive and unquantifiable factors.

Except for a few specific problems there did not seem to be great concern about line generalization among DMA cartographers. They were worried that contours crossing drainage show appropriate turn back, that deleted drainage patterns be supported by the contours, and that terrain features be correctly represented. They did not seem to be concerned particularly about oversmoothing or standardization between pull-ups.

This apparent lack of concern may be because line generalization is easy to perform intuitively, or it may simply be that due to the difficulty in stating and applying standards for generalization. Compilers are not as aware of the possibilities for error as they are with the feature displacement problem.

This does not mean that line generalization is a simple automation problem. Often those tasks which people find easiest to perform are the hardest to automate. It does indicate that the choice of the "best" algorithm for line generalization may not be as important as guaranteeing that the contours are tied to drainage in the correct manner, that features are emphasized, and that map accuracy standards are maintained.

3.3.2 Feature Displacement

Significant features and feature hierarchies vary greatly depending on product and facility. One consequence of this variation is that any general purpose displacement algorithm must accept feature hierarchies as parameters, rather than being based on a fixed set of priorities.

Present DMA specifications for handling feature displacement are not fully adequate. However, according to survey responses less than 10% of the problems are not currently covered by some sort of guidelines. Also the survey responses indicated that the overwhelming majority require dealing with only two features at one time. It is difficult to visualize how complete specifications could be produced to handle every case, since any displacement problem increases in complexity at a geometric rate as features are added to an area in question.

Attempting to develop such complete specifications almost certainly demands an iterative process beginning with the listing of known guidelines and using feedback from cartographers to expand the rules as those guidelines are used in ongoing work. Since the majority of problems are covered by specifications it seems that development of a displacement algorithm could be initiated without using much in the way of supplementary information and achieve useful results even though it did not fully resolve all difficult situations.

At a number of sites cartographers felt justified in dynamically changing specifications to handle extremely difficult displacement problems. Both non-standard symbology and modified feature classifications were mentioned as options. For example, in a case where a primary road must pass between two features which cannot be moved and the road will not fit using the standard line weight, it is sometimes permissible to reclassify the primary road as a secondary road in that region if the resulting reduction in line weight will resolve the conflict.

3.3.3 Limitations of Current Procedures

ZYCOR observed several cases in which symbols and amounts of generalization were noticeably different between pull-ups generated by different cartographers for the same map. The standardization available from an automated system would reduce this problem. It would also eliminate the difficulty of tying information on one pull-up to that on adjacent pull-ups.

Several cartographers suggested that there are conceptual limits on the ability to generalize if the ratio between the input scale and the output scale is too large. This limitation implies multistep manual compilation procedures which can lead to increased error in map production.

It would not be expected that an automated system would have intrinsic scale reduction limitations. However, if human beings do have difficulty dealing with major scale changes, a cartographer's ability to interact with an automated system may also be limited. For example, if the system marks a complex area for human processing the cartographer may have difficulty resolving the problem if the output is much reduced from the sources.

Use of zoom capabilities and variable line weights on sophisticated graphics terminals may reduce this problem.

3.3.4 Quality Control

Quality control is a complex process at DMA. It may be performed at a number of different stages and by different individuals. The information content on maps is so large that quality control on final products is sometimes a matter of sampling only the most densely symbolized areas and assuming that the results from that check can be extrapolated to the entire map.

Automation of generalization and feature displacement would affect this process in two ways. First it would provide a level of standardization which should make error detection easier. Second it might be able to automatically indicate those areas of the map which had created the most difficulty in compilation and which therefore demand the most serious study by a human inspector.

4.0 LINE GENERALIZATION ALGORITHMS

During the last 15 or 20 years numerous approaches to the generalization of linear cartographic data have been developed. This chapter presents an overview of the relevant work which has been published in the cartographic and computer science literature.

In order to present this material in an orderly fashion it is helpful to assign algorithms to categories. Unfortunately, the "correct" classification method is not obvious. Several are suggested in the literature. Poiker (1973) divides algorithms generally into three classes: those which eliminate points along the line; those which approximate the line with a mathematical function; and those that delete specific cartographic features represented by the line. Marino (1979) suggests a more fundamental division; those which eliminate points and those which select points. In this report, ZYCOR has attempted to make a classification according to the predominate mathematical techniques used in a particular algorithm and has derived nine classes of line generalization techniques including selection, low pass filtering, angle detection, DEM smoothing, tolerance bands, point relaxation, domain transformation, mathematical fitting, and epsilon filtering.

A related difficulty is deciding how much detail to provide within specific categories. It is not possible to provide information on every variation in implementation of all line generalization techniques. Emphasis is given to algorithms which are well established in the cartographic community; which provide representative overviews of general ideas; or which provide interesting technical approaches to the problem. In many cases, the algorithms are complex and require a great deal of mathematical notation to fully describe them. In these cases, the basic ideas are presented. Reference to original sources will be necessary if it is desired to understand fully how these particular algorithms are implemented.

Another factor should be noted at this time. These algorithms will be part of a map compilation process which must give attention to complex data base and topological considerations. For example, many points along a digitized cartographic feature may not be free to move because they form nodes in a spatial data base with ties to other features. This is reflected in the figures in this chapter which always show the end points of curves held fixed. In practice this may effect algorithm performance by restricting the length of features which may be presented for generalization. Furthermore, when generalizing any feature, care will have to be taken to avoid upsetting fundamental topological relationships. For a straightforward example, two linear cartographic features which do not cross before generalization must not cross afterwards. In a complete system data base aspects may actually be more computationally complex than the generalization algorithms.

4.1 SELECTION

Apparently the oldest and certainly the simplest algorithms for line generalization are based on "arbitrary" selection of points according to procedures which are independent of the cartographic representation of the data. For these algorithms the rules for selecting points are dependent only on the simplest of mathematical relationships.

The most straightforward implementation is completely arbitrary: every n-th point is retained to represent the line. A smaller value of n implies less generalization and a larger value more. Such a procedure is natural in a case where it is known that linear data has been uniformly over-sampled. This is often the case for the data produced by automated digitization equipment (Tobler, 1966; Rhind, 1973; Robinson, 1978). Figure 4.1 shows an example of the application of this algorithm for the case n=2.

Another approach involves filtering based not on the number of points encountered but on the distance traversed. Here the arc length between successive points in the original data set is calculated. The data is filtered by deleting every point which follows a selected point within a tolerance distance or by selecting the next point closest to the tolerance distance.An extension may be made by specifying a minimum acceptable separation between retained points. A case in which this method is superior to the simple n-th point selection is provided when a human digitizer has moved at different speeds over regions of varying complexity while the sampling logic of the digitizer continued to work at a fixed rate. This method provides more control than simple selection but is still likely to oversample straight segments of a curve and undersample very complex regions. Figure 4.2 shows an example of the application of this algorithm.

A more complex approach can make use of probability theory. Here the points to be eliminated may be chosen in a pseudo random manner. For example, generation of a pseudo-random sequence of numbers between 0 and 1 while traversing the data, and throwing out every digitized point corresponding to a random variable larger than .5 would statistically have the same effect as throwing out every other digitized point. This method decreases dependence on starting points and guarantees that an unfortunate choice of parameters does not result in points being eliminated which coincide with some fundamental frequencies in the data.

These algorithms are computationally efficient. They provide an easily controllable reduction in data size. However

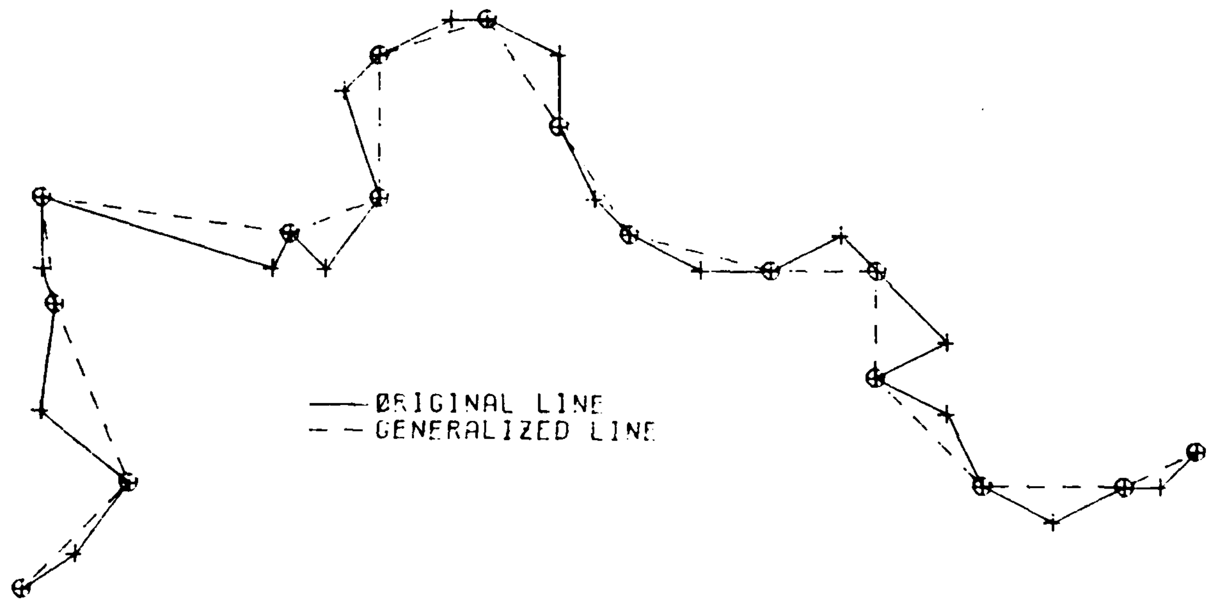


Figure 4.1. Simple Selection Algorithm, N=2

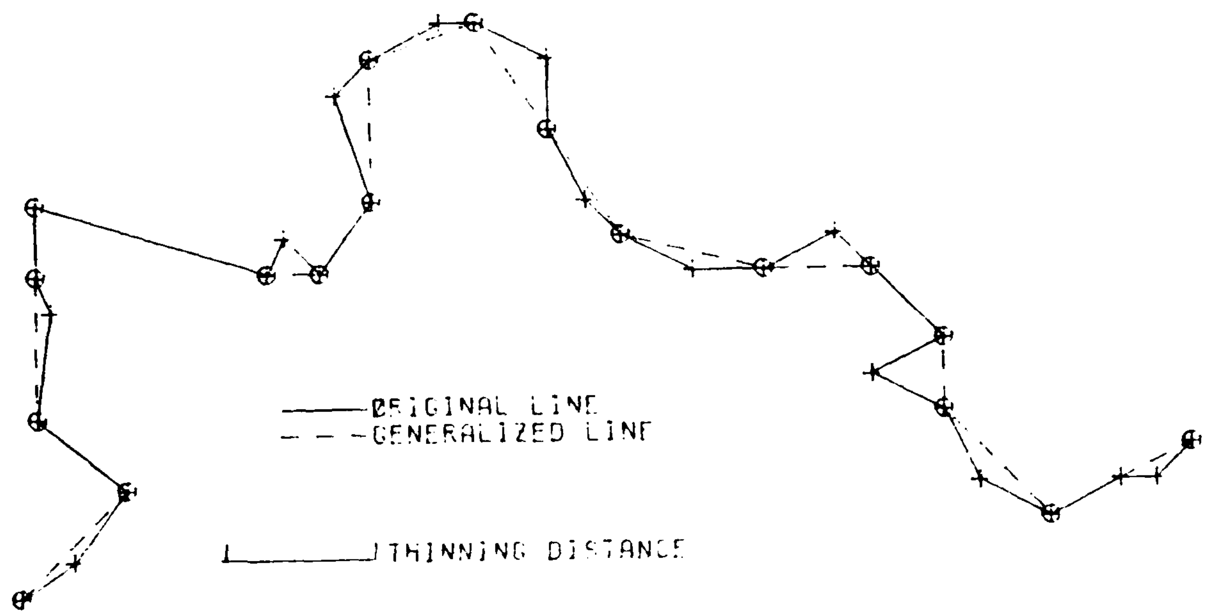


Figure 4.2. Distance Selection Algorithm. The last point within the distance tolerance is chosen.

they can accurately represent the character of the line only if the data is very dense and the sampling interval is small. They may easily miss significant features on curves or corners and over represent straight line segments.

Furthermore, these algorithms have no relationship to cartographic rules which can be used to guide generalization. This is important since they can actually demand considerable effort on the part of the user, upon whom the entire responsibility for maintaining the cartographic meaning of the line must fall through his choice of a thinning parameter which has no obvious relationship to line character or meaning. Valuable time may be spent iterating to find the correct parameters in any given situation.

4.2 LOW PASS FILTERING

Another early algorithm which has been used in line generalization produces smoothing through averaging. This approach obtains the points of the generalized line by computing the weighted or unweighted means of the coordinate values of a set of sequential points. The amount of overlap between successive data collections determines how much data reduction takes place. The more overlap between the adjacent groups of points the higher the accuracy of the new data. This higher accuracy is obtained at the price of greater data retention. (Tobler, 1966; Holloway, 1958) Figure 4.3 shows a simple case with overlap and 3 points averaged.

The width of the window to be used in smoothing may be either determined by specifying a number of points to include or by specifying an arc length width, using all points which fall within the window. The wider the window the more smoothing that takes place. Also, the more evenly apportioned the various weights are, the greater will be the smoothing. Theoretically,

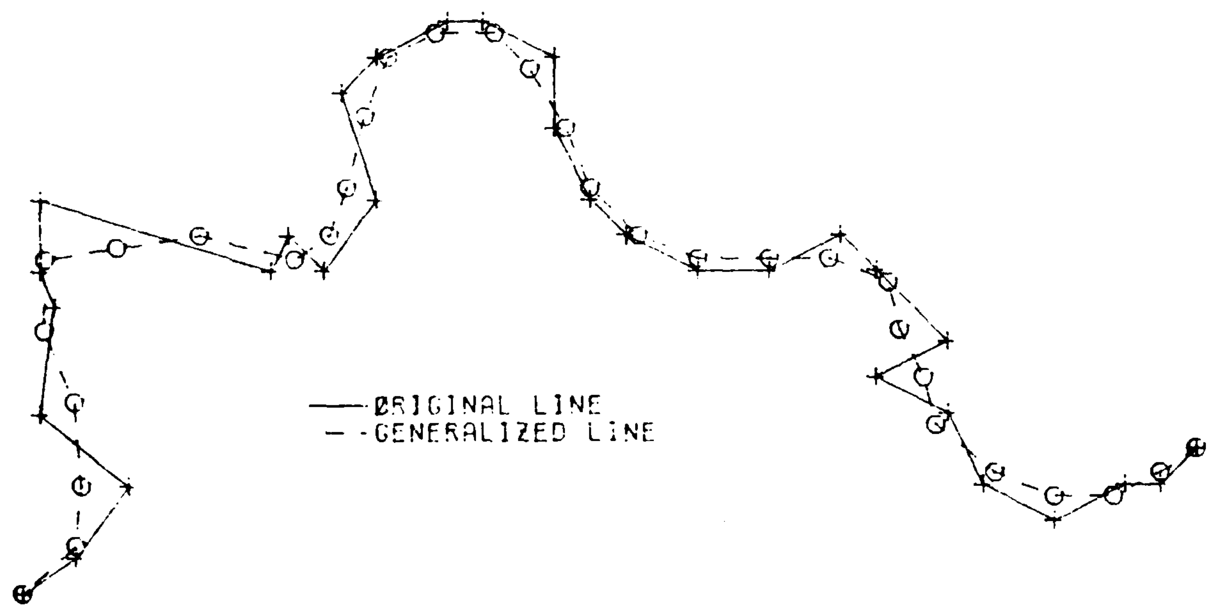


Figure 4.3. Moving Average. Size of Window is 3.

the selection of weights should be made on the amount of autocorrelation in the data. (Robinson, et al, 1978)

Gottschalk (1973,1974) presents frequency domain procedures for determining the width of the optimal window based on the width of the power spectrums of the two coordinates of the line. Such techniques are not independent of the coordinate system being used or invariant with respect to rotations of the data.

Robinson, et al provides several examples of weighting functions for equally spaced vertices. Jancaitis and Junkins (1973) suggest an approach based on "weighted centroid smoothing" with weights determined by distance from a central location. Obviously there are many reasonable approaches to deriving "good" weighting functions. In general, any smoothing function using positive weights will cause closed convex features to shrink. Opheim (1981) suggests the use of an approximation to the ideal low pass filter function which reduces this problem by the introduction of negative weights.

These algorithms share certain disadvantages of the simple selection algorithms discussed above. They place much responsibility on the user to define parameters. Furthermore, they always smooth out extreme points and reduce angularity in a line, exactly the features which are often identified as "significant."

On the other hand, low pass filters are designed to remove high frequency information. They do this well. In a situation in which cartographic goals are such that the general trends of the data are more important than local detail, these algorithms may be very useful.

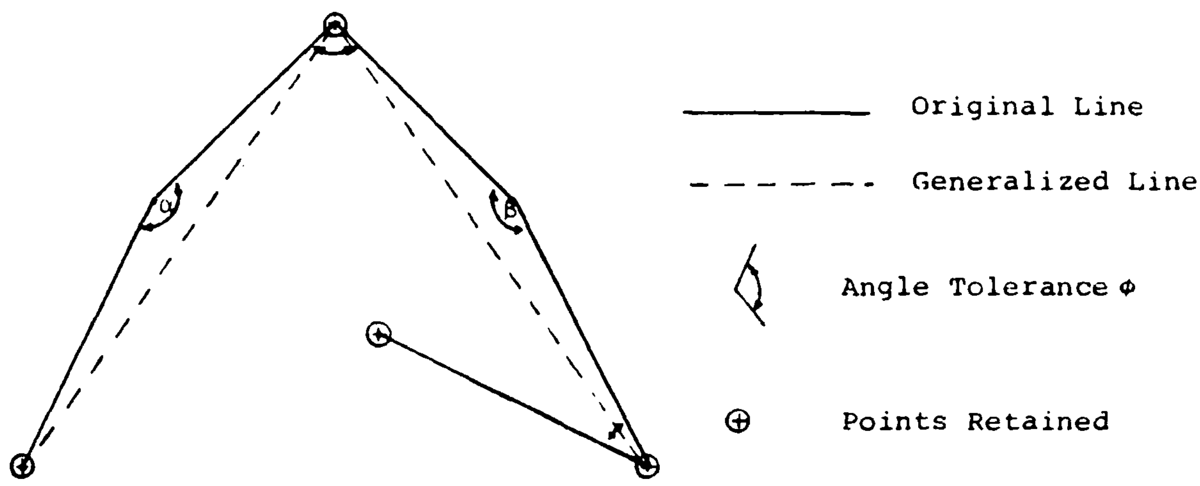
4.3 ANGLE SELECTION

These algorithms attempt to locate and retain points along a digitized curve which represent significant changes in direction. There are numerous simple approaches which focus on individual vertices. One consists of calculating the angle between the vectors joining a curve vertex and its preceding and succeeding points. If the angle exceeds a predetermined threshold the middle point is retained; if not, it is deleted. Another algorithm creates a 'field of view' around a line connecting the first and second points. The field of view is defined by a preset tolerance angle and the third point is retained if it falls outside this field of view. Since these two algorithms do not take distances into account they can prescribe the same performance for data with significant visual differences. These algorithms are shown in Figure 4.4. There are straightforward extensions which take account of distance by requiring points to be selected at a minimum distance interval even where the angular change is not large.

More sophisticated approaches involve iterative techniques which compare points and attempt to select those with the most information content based on angle measurement and other characteristics. Often these algorithms are called "dominant point" algorithms since they attempt to segment the input curve into arcs which are dominated by a single point. There are essentially two approaches to detecting these dominant points:

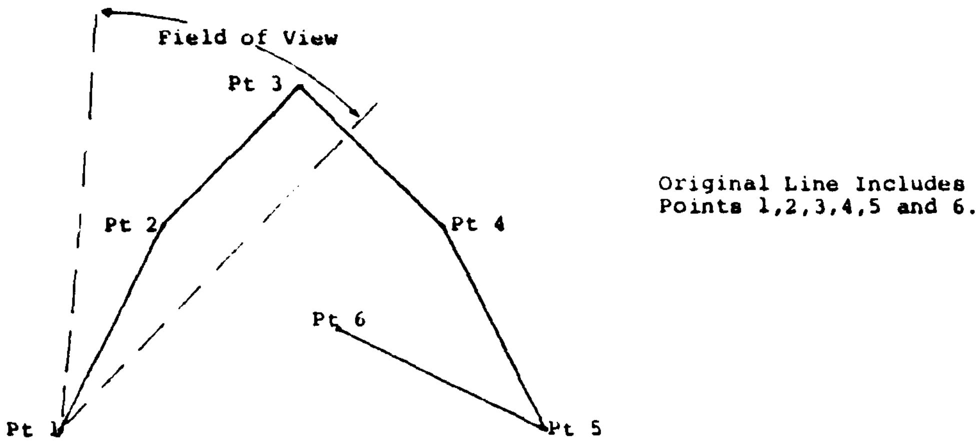
1. Start with two arbitrary points and iteratively include more points based on some criterion until a reasonable approximation is obtained.

2. Consider all points on the curve and then iteratively remove points based on some criterion until a reasonable approximation is obtained (Sankar and Sharma, 1978)

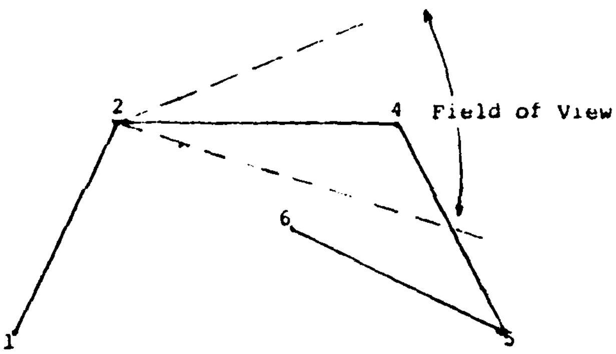


NOTE: Points with angles α and β are eliminated because α>Ø and β>Ø

Figure 4.4A. Angle Calculation at Each Point



Step 1: Point 3 is inside the field of view, so it is eliminated.



Step 2: Point 5 is included, it falls outside the field of view.

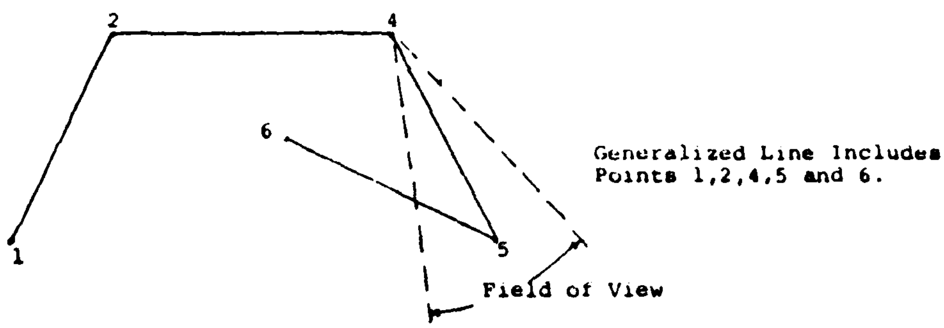


Figure 4.4B. Angle Calculation with a Field of View

The first approach includes methods which are called tolerance band algorithms in cartography and which are described in section 4.5. There are a number of different approaches to the second method, each based on heuristic techniques.

Rosenberg (1972) presented an algorithm designed to work on convex sets. His algorithm effectively divides a digitized curve into segments which are dominated by a single point. These dominant points and segments have certain characteristics including:

1) The total angle change from start to finish of the segment is not greater than 90 degrees.

2) The break points between segments tends to minimize the ratio of arc length distance to chord as measured from dominant points of the two adjacent segments to all intermediate points.

3) The angle defined by the break points of a finally chosen segment and its dominant point is smaller than that provided by other possible segmentations which would have dominant points within the range of that segment.

As described by Rosenberg the segments for dominant points grow by "gobbling" up less significant points and ranges. More details are available in Rosenberg's paper. A recent paper by Rutkowski (1981) questions the use of arc to chord ratios in curve segmentation. Rutkowski shows examples in which it yields results which are not perceptually plausible.

Rosenfeld (1973, 1975) describes a technique which focuses strictly on angle calculations. At each point on the input curve a sequence of angles is defined by vectors constructed from the point to the n-th points preceding and succeeding the point as shown in Figure 4.5.

For a given vertex i on the input curve the cosines form a sequence

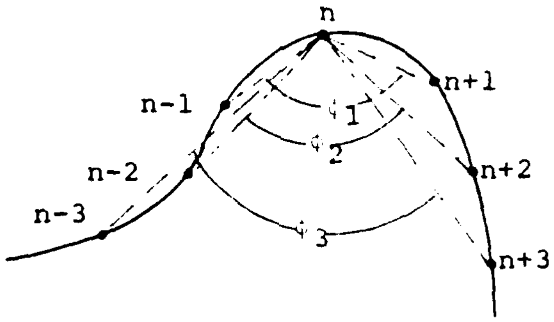


Figure 4.5. Successive Angle Calculation

$$\cos_i(1), \cos_i(2), \ldots, \cos_i(m)$$

The "best size" for vertex i is h(i) such that

$$\cos_i(m) < \cos_i(m-1) < \ldots < \cos_i(h(i)) \geq \cos_i(h(i)-1)$$

Then the vertex i is defined as an "angle" point if $\cos_i{}^{h(i)}$ is a local maximum in the sense that

$$|i-j| \leq h(j)/2 \text{ implies } \cos_i{}^{h(i)} \geq \cos_j{}^{h(j)}$$

Rosenfeld suggests using a prespecified fraction of the number of points in the curve as the length of these sequences (e.g. $m=N/10$), however this does not account for variations in sampling rates or in the length of line segments. This algorithm is highly dependent on the parameter m. For a fixed value of m the angle estimates at a point may not agree with its perceived size. Also, if there are two "significant" angles within m vertices of each other, the sharper angle will suppress the other.

Rosenfield (1975) suggested an improved algorithm which reduced this dependence on m slightly by replacing $\cos_i{}^{(k)}$ by an average of $\cos_i{}^{(j)}$ for $j \leq k$. Davis (1977) started with the same basic algorithm but developed a complex search procedure which associates a set $\cos_i{}^{(k)}$, $j_1 < k < j_2$ with each vertex and searches through that set for fixed values of k over all choices of i in order to guarantee detection of all possible local maximums. In this algorithm a point (angle) is stripped off its domain only if some smaller, yet comparable domain (in significance) is contained within it.

These angle algorithms are strongly motivated by perception research. They attempt to find directly those points of high angularity. They can produce unnatural outputs with many sharp corners. Also, their use is not particularly intuitive. A direct tie between angularity and a particular degree of generalization is not clear. Furthermore the rules utilized tend to be highly heuristical and a method of choosing a best algorithm or the correct control parameters for a particular algorithm is not obvious.

4.4 DEM SMOOTHING

Generalization should consider features to be as important if not more important than lines. Much of the emphasis on linear data is due to the fact that algorithms which work on sequential data are generally easier to program; linear data is a natural storage format; and, at least physically, cartographers tend to process one line of data at a time during generalization.

To develop a truly general operator based on linear data would be difficult. For example, keeping track of adjacent contour strings when they may arbitrarily separate and move together over terrain would be next to impossible. Simply detecting crossing of line data is a computationally difficult procedure. However, contours are only one method of representing terrain. Digital elevation models (DEM's) provide a computationally more versatile representation which allow for easier access to information at particular locations and at relative locations.

Using DEM's as a data base which can be smoothed by area type filtering and then contoured as a method of generalization of contour data has been suggested several times (Bassett, 1972; Loon, 1978; Lichtner, 1979). Loon describes several DEM smoothing filters including least squares collocation algorithms and local 9 point convolution filters. The shape of the filters may be modified in order to adapt to the data. Table 4.1 shows the weights used in Loons "nines filter" based on a Gaussian shaped smoothing function

$$C(d)=\exp(-a^2d^2)$$

with a = .83.

TABLE 4.1

.063	.125	.063
.125	.249	.125
.063	.125	.063

In the above case a is a free parameter which can be adjusted to take into account the covariance properties of the data, and d measures distance from the center node. Special filters need to be calculated for the edges and corners of the grid.

Davis, et al. (1982) have suggested the use of 13 point biharmonic and 5 point Laplacian filters for the same purpose. The biharmonic operator is particularly interesting since it functions to reduce surface curvature (Briggs, 1974). Curvature is defined at each point on the grid by the quantity

$$(Z_{i-1,j} + Z_{i,j-1} + Z_{i,j+1} + Z_{i+1,j} - 4Z_{i,j})^2.$$

Attempting to simultaneously minimize this value for all nodes on a large DEM results in a matrix algebra problem outside of the capabilities of current computer systems. Thus, a recursive method of applying the filter is required. This recursive approach is advantageous since it is not necessarily minimum curvature which is the goal, but rather a certain level of DEM smoothness which may be related to contour generalization.The affect on contour roughness of applying the biharmonic filter to a data set is shown in Figures 4.6 through 4.8.

There are numerous other possible smoothing algorithms (Allam, 1978). Harbaugh and Merriam (1968) discuss Fourier and least squares techniques including special methods of dealing with DEM edge conditions during Fourier smoothing.

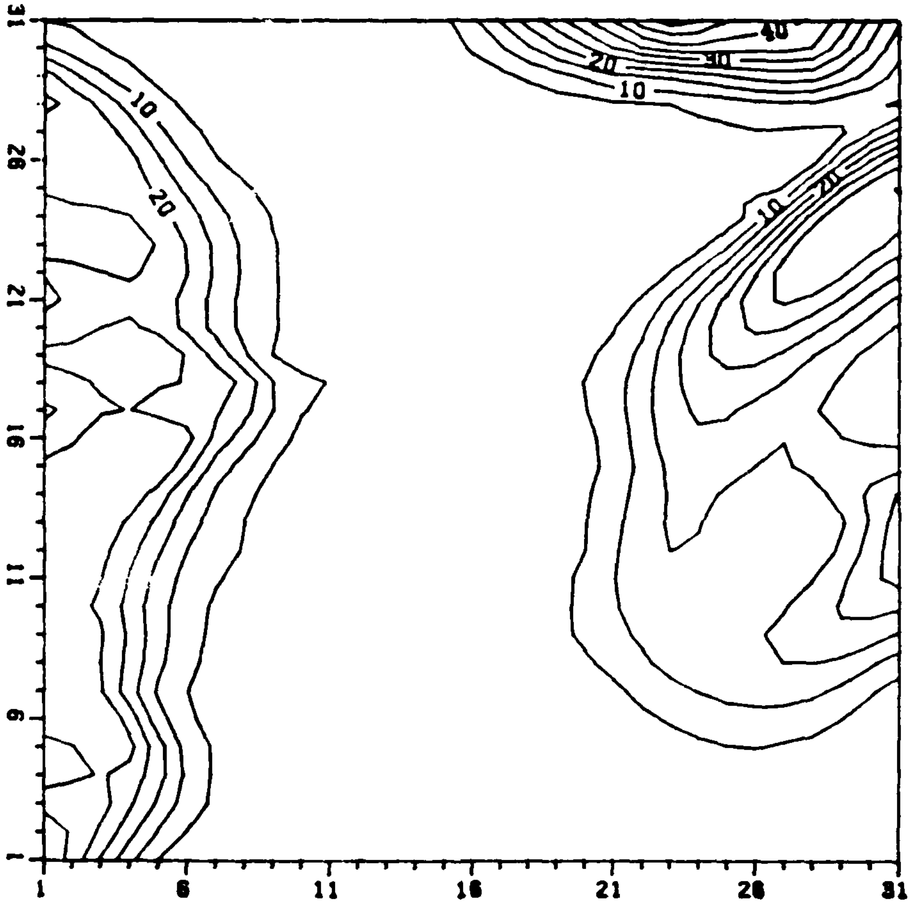


Figure 4.6. Contoured DEM

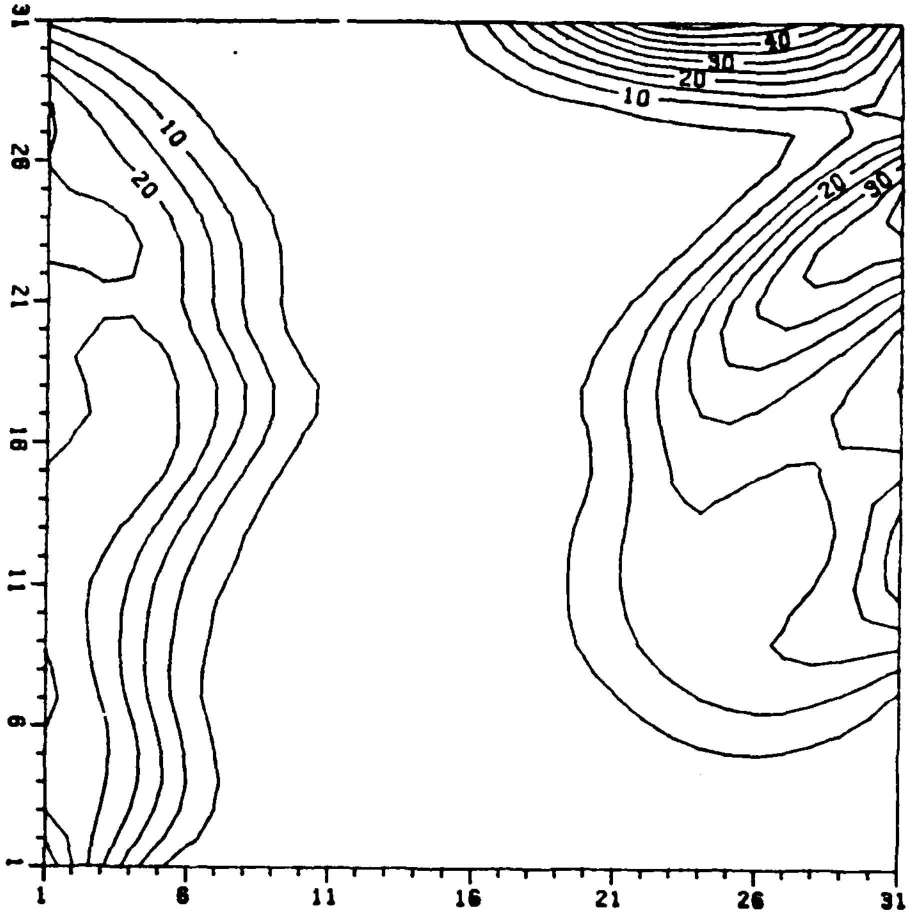


Figure 4.7. Contoured DEM After 1 Biharmonic Filter Pass

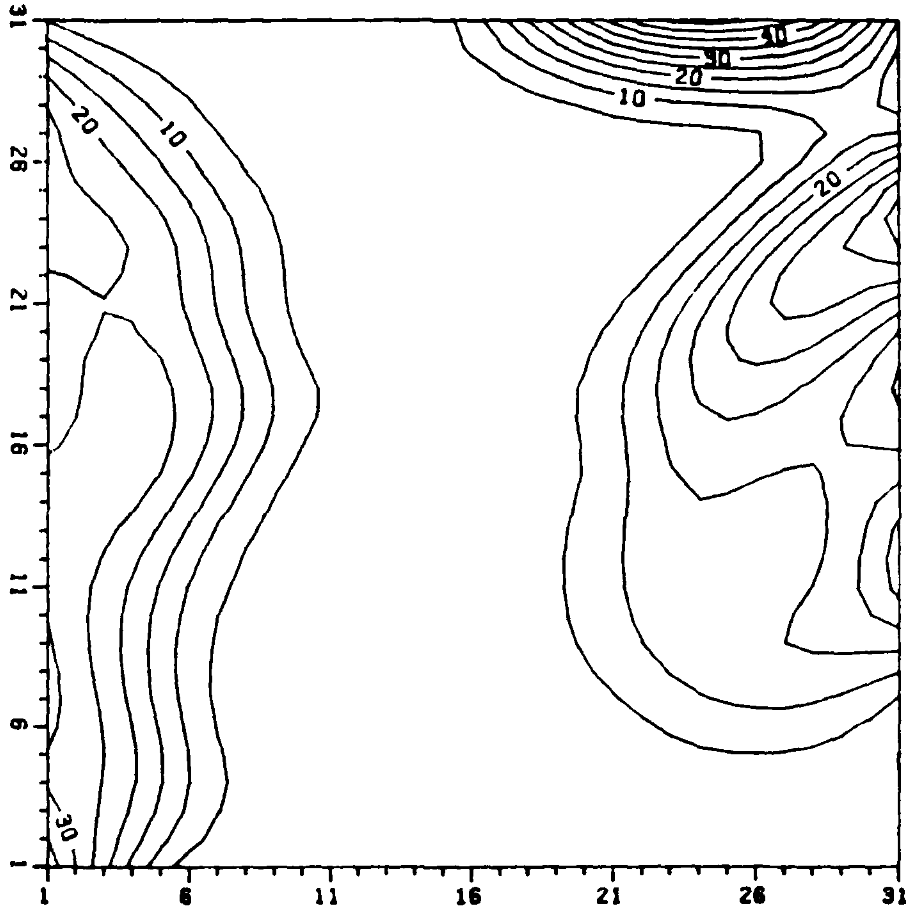


Figure 4.8. Contoured DEM After 2 Biharmonic Filter Passes

DEM smoothing for generalization has a number of advantages besides making use of area operators. By using easily defined surfaces which put upper and lower constraints on changes on all elevations it is possible to define rigorously the amount of error created in the generalization process.

It may be possible to tie features such as drains, spot elevations, and roads into the DEM or into asosciated contouring software. Jancaitis and Junkins (1973) suggest mathematical models to be used for that purpose.

Using DEM as a basis for contour generation means that any desired contour interval may be supported. This is not true when using digitized contour data directly.

There are also significant unsolved problems with using this approach to contour generalization. Most importantly it requires the existence of contouring software which can produce specification quality output. Current algorithms do not yet satisfy this demand. Nor have methods for imbedding a wide range of features into the DEM or the contouring software been demonstrated. Even if the necessary algorithms exist, it is not clear that existing DMA DEM resolution is sufficient to support large scale mapping requirements.

Furthermore, the "correct" methods of DEM smoothing for generalization purposes is not clear. A straightforward connection between DEM roughness and perceived contour character does not currently exist.

A related problem is that the use of grid smoothing for contour generalization may not be easy to control. The wide range of measures of DEM roughness available may provide sufficient guidance.

4.5 TOLERANCE BANDS

A class of algorithms which establish bands or areas which form templates for inclusion or exclusion of points from the original data set are called tolerance band algorithms.

4.5.1 Hysteresis Filtering

The concept used in all tolerance band algorithms is similar to a filtering technique used in digital signal processing called hysteresis smoothing (Duda and Heart, 1973). It is applied to a time varying signal as shown in Figure 4.9. A vertical window is defined and centered at the beginning of the data. The window is then moved horizontally until one of its ends touches the data. The window continues to move horizontally but is also "pulled" up or down by the data touching the end of the window. When the data no longer touches the window ends, it reverts to a purely horizontal movement. The smoothed output is produced by tracking the center of the window. This approach is computationally simple and easily controllable. Peaks and valleys smaller than the size of the window are removed. However "significant" fluctuations are retained.

The basic algorithm has a serious drawback in that the smoothed output has been shifted to the right. This is caused by tracking the center of the window while the ends of the window are detecting and following the data. Ehrich (1978) has suggested a modification which keeps one end of the window on the data at all times and creates the new curve by tracking the window boundaries rather than its center. His modification prevents the data shift at the price of requiring two passes through the data.

Hysteresis filtering assumes that there is a well defined coordinate system to which the window may always be refer-

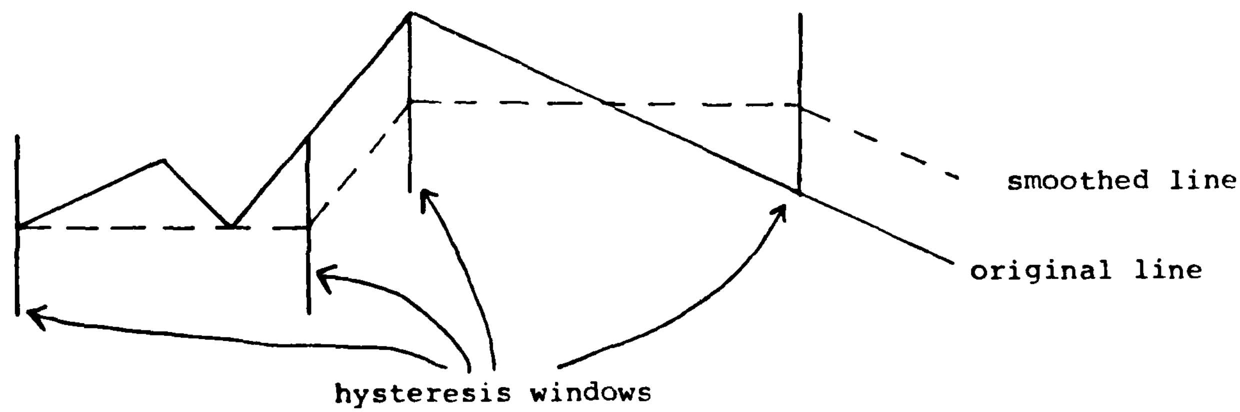


Figure 4.9. Hysteresis Smoothing

enced. This is not true for linear cartographic data which moves in an uncontrolled manner over a map. Although it is not clear that any of the tolerance band algorithms used in cartography were developed with this filtering method in mind, they have many of its properties along with a continually changing basis with respect to which something like the hysteresis window may be referenced.

4.5.2 Simple Local Tolerance Band Algorithms

Lang (1969) suggests an algorithm in which a starting point is sequentially connected to subsequent points to create reference lines. The perpendicular distance from each intermediate point to the current reference line is computed. When the distance for an intermediate point exceeds a tolerance, the last previous reference line becomes part of the generalized curve and the algorithm is continued starting from the point which defined that reference line. This is shown in Figure 4.10. Recently Williams (1978) has presented an extremely efficient implementation of this algorithm using polar coordinates.

It is possible to restrict the above approach to only three points from the original data, with the middle point referenced to the line joining the two end points. Another modification can involve using the last selected point and the next sequential point to define a base line. Subsequent points are measured with reference to this line and the first to exceed a predetermined distance is selected for inclusion in the smoothed curve.

4.5.3 Global Tolerance Band Algorithms

A decade ago several authors developed a tolerance or band approach which considers the entire line during processing. This mimics the global views of a trained cartographer in observing the entire line during generalization.

Ramer (1972) had suggested a global algorithm which works as follows: the first and last points are connected by a straight line. A band of fixed width is defined around this line and all intermediate points are checked to see where they fall in relationship to the band. If they all are within the band then the assumption is that the single straight line satisfactorily reflects the data and all the intermediate points are eliminated. If some fall outside the band then the point which falls farthest from the segment is considered significant and the original single segment is broken into two: one joining the first point to the newly selected point and the other the newly selected point to the old second point. This selection process is shown in Figure 4.11. The algorithm is then repeated using as inputs the two new segments using exactly the same logic. At the termination of the algorithm the generalized line is then considered to be made up of the first and last point plus all the intermediate points detected by the algorithm. The sequence of curves in Figures 4.12 through 4.15 shows the successive application of this algorithm to a given line.

Douglas and Poiker (1973) suggest two implementations of this concept which are algorithmically superior. In the first method the first point on the line is defined as an anchor and the last as a floating point. These two points are used to define a straight line segment. 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 the distance to that point is larger than the maximum tolerance distance, the point lying farthest away becomes the new floating point. As the cycle is repeated the floating point advances toward the anchor. When the

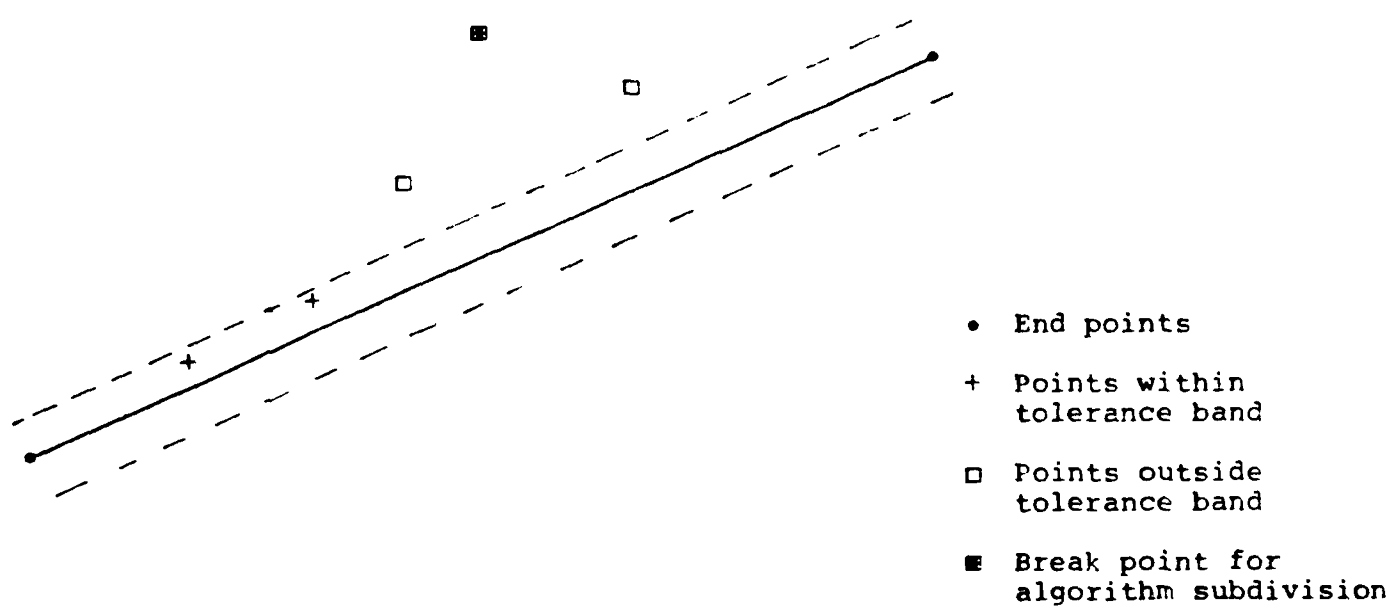


Figure 4.11. Tolerance Band and Point Selection

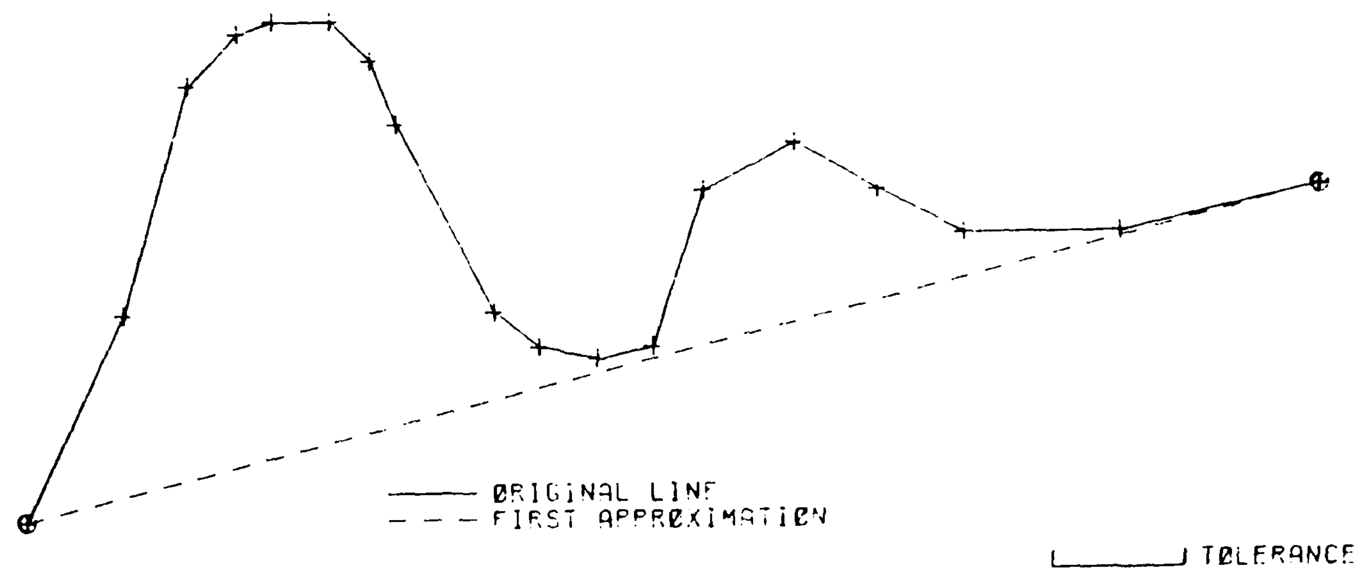


Figure 4.12. Original Line and First Approximation Using the Ramer Algorithm

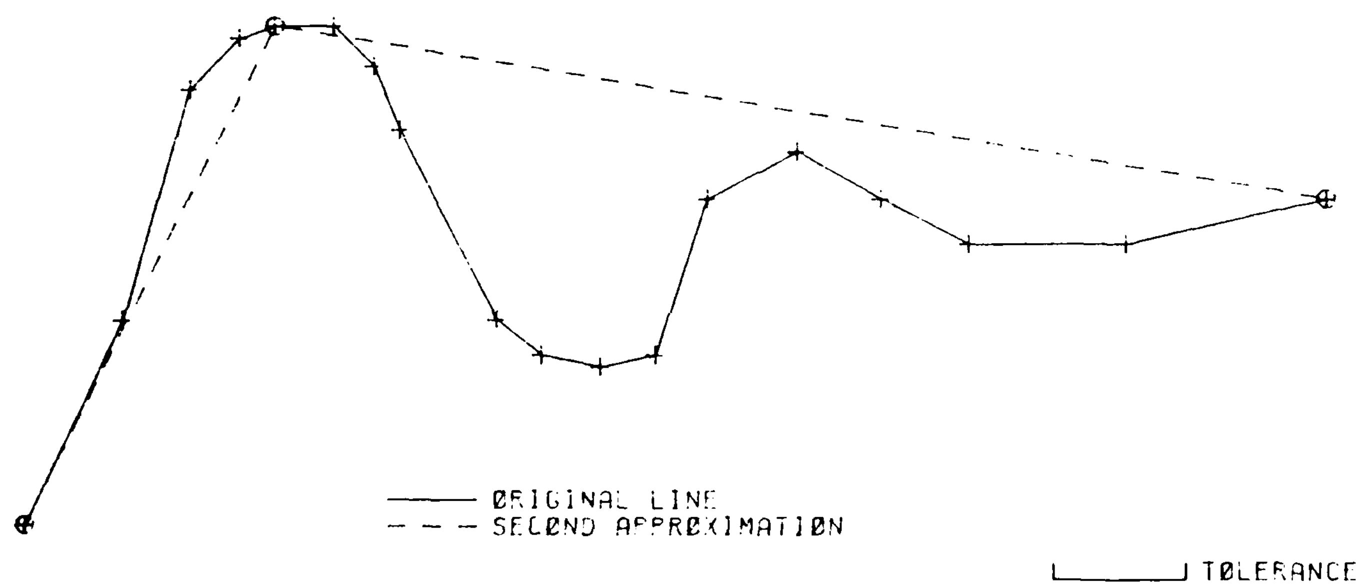


Figure 4.13. Original Line and Second Approximation Using the Ramer Algorithm

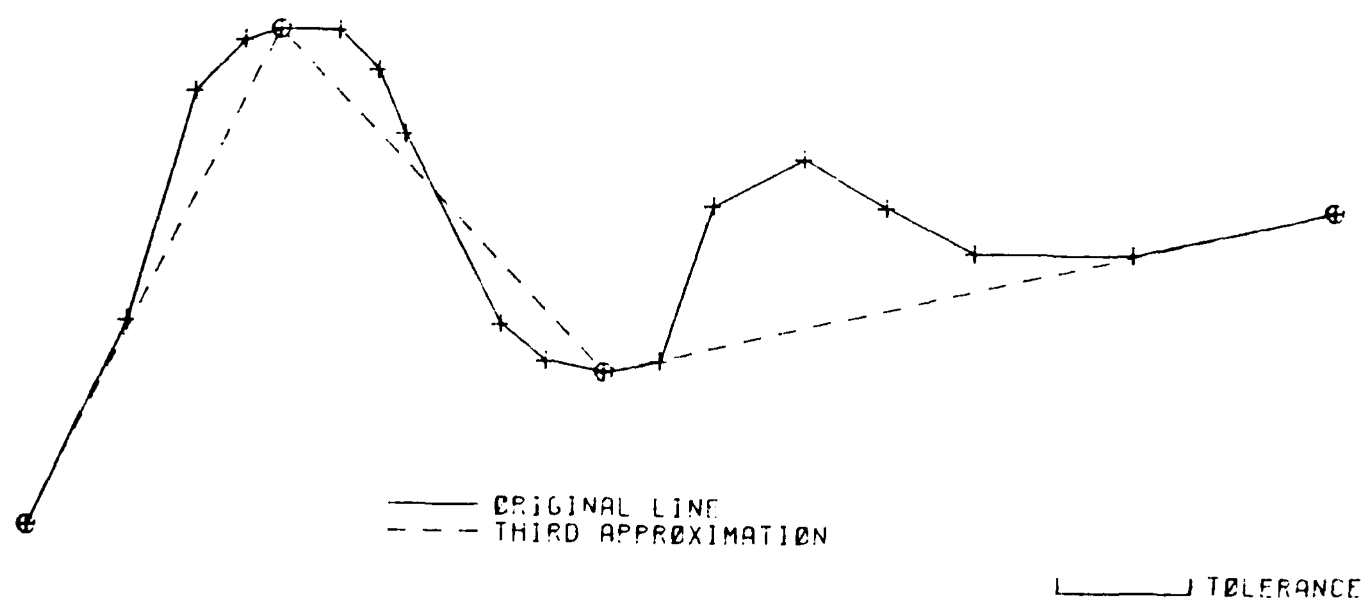


Figure 4.14. Original Line and Third Approximation Using the Ramer Algorithm

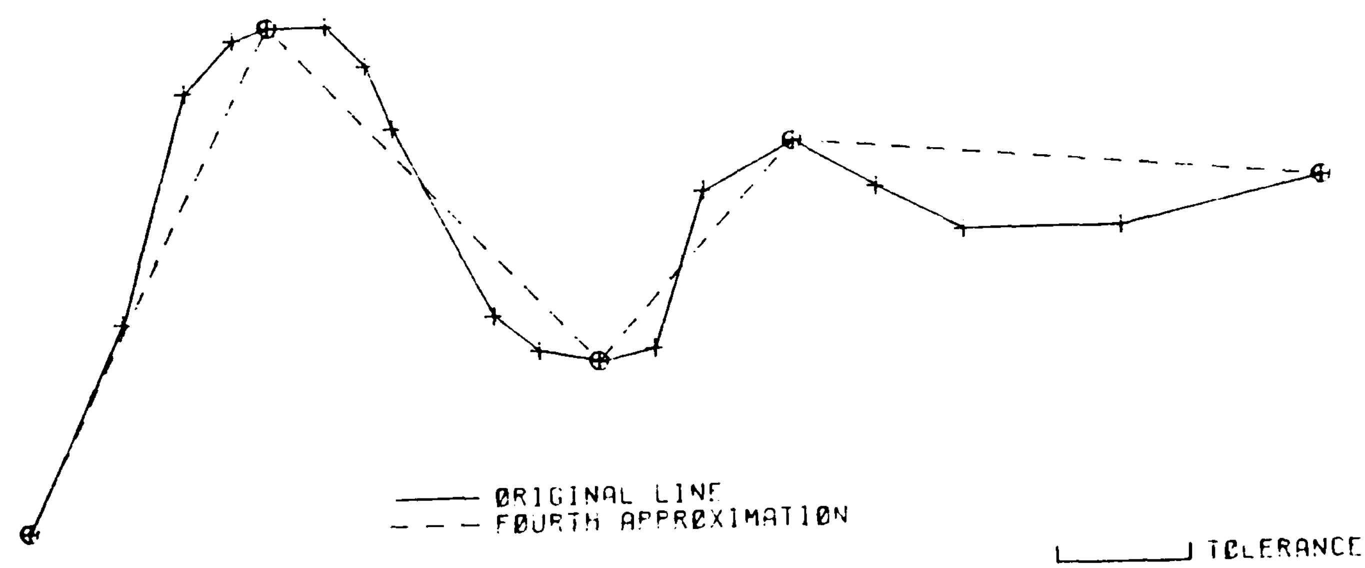


Figure 4.15. Original Line and Fourth Approximation Using the Ramer Algorithm

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. All points which are assigned as anchor points comprise the generalized line.

In the second method the same basic approach is used except that all the points which are assigned as floaters are recorded during an iteration and stored in a stack. After the anchor is moved to the floating point, the new floating point is selected from the top of the stack instead of being reassigned to the end of the line. This significantly reduces the number of distance calculations which must be made. This method selects more points but is reported to use only 1/20 of the computer resources.

The amount of smoothing obtained is a function of the amount of detail in the original data and the width of the band. The width of the band is the tolerance which gives these algorithms their name. It is a parameter which is easy to understand and has an intuitive meaning which may be related to the pen widths used in the creation of pull ups.

4.5.4 Localized Implementation of Douglas Poiker Concepts

Reumann and Witkam (1974) have developed a method which makes direct use of the tolerance bands concept but in a local utilization. A band defined by two parallel lines is created with the lines sloping in the direction of the tangent to the original curve at the last included point as shown in Figure 4.16. A point inserted where the curve crosses out of the band or the last input point contained within the band is selected for retention. The algorithm is then repeated based on the last point and its tangent. Figure 4.16 shows the results of applying this algorithm to a curve.

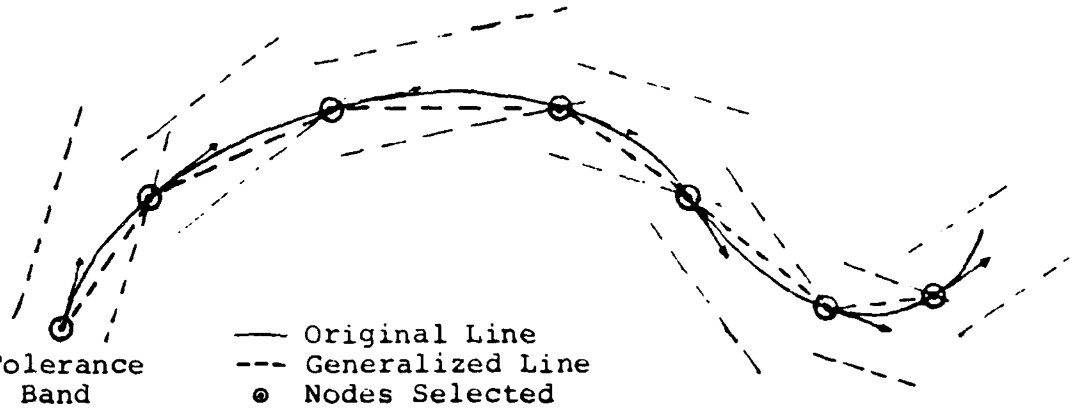
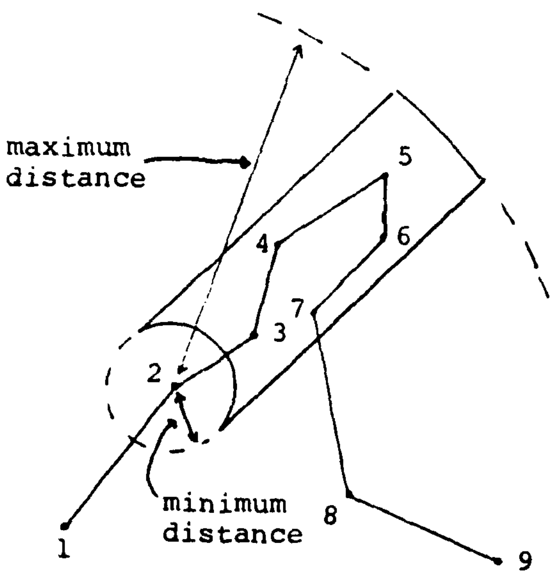


Figure 4.16. Reumann-Witkam Algorithm



NOTE: Points 3,4,5 and 6 can be eliminated since they fall inside the band.

Point 7 becomes part of the generalized line. It is the last point to fall inside the search region.

Figure 4.17. Ophelm Search Band

Opheim (1981) presents several modifications to the Reuman and Witkam method. First he suggests adding a minimum and maximum distance to the search region, with the requirements for point selection being that the selected point not be closer to the current point than the minimum distance and farther than the maximum distance. He also suggests applying the Douglas Poiker algorithm to the line segment between the current selected point and the new selected point to detect small turn backs in the data. Figure 4.17 shows a search band as suggested by Opheim and a possible turn back situation which would be detected by use of the Douglas Poiker approach.

Both the Reumann and Witkam algorithm and the Opheim algorithm require calculation of the tangent to a dititized curve. Various methods have been developed. to approximate the standard discrete formula given by $T=(x',y')$ where

$$x' = \frac{1}{d_1}(x_2-x_1) + \frac{d_1}{d_1+d_2}\left(\frac{1}{d_2}(x_3-x_2) - \frac{1}{d_1}(x_2-x_1)\right)$$

$$y' = \frac{1}{d_1}(y_2-y_1) + \frac{d_1}{d_1+d_2}\left(\frac{1}{d_2}(y_3-y_2) - \frac{1}{d_1}(y_2-y_1)\right)$$

Another local method which requires distance tolerances is provided by Dettori and Falcidiendo (1979). This method works as follows:

1) Start with the first two sequential points.
2) Add the next sequential point.
3) Compute the minimum convex hull (polygon) which contains this set of points. (This convex hull is defined by the smallest subset of the vertices which define a polygon containing all vertices. Efficient algorthims exist for calculating this subset.)
4) For each side of the hull, compute the distance to all vertices. If a side exists for which all vertices are within a predefined tolerance then iterate the algorithm by adding

another point and starting at hull creation. If not, the last point for which success was obtained is selected as being included; all intermediate points before it thrown out. The algorithm now begins at step 2 with the last included point.

Figure 4.18 shows the sequence of convex hulls created for a curve and the resulting generalized curve. An advantage of this algorithm in certain applications is that it automatically eliminates spikes which are not eliminated by other methods.

4.5.5 Critical Points and Tolerance Band Algorithms

These tolerance band algorithms may choose points of low curvature for inclusion in the generalized curve. Also, while they come close, they do not necessarily choose exactly the points of highest angularity. Liao (1981) presents a post processing algorithm which uses the original data and the generalized curve to eliminate unneeded points and move selected points to their correct location. In his algorithm, adjacent pairs of segments in the generalized curve are processed together. Using the distance measurement from the original algorithm and using the original data an attempt is made to replace the two segments with a single line. If this is impossible then the point which provided the maximum deviation is used as the new joint between the two segments. This processing is repeated until none of the points in the generalized curve may be removed or moved.

4.6 POINT RELAXATION METHODS

A number of algorithms have been developed based on straight line approximations constrained to pass through circles centered at vertices of the input curve. The position of each point is allowed to "relax" or move away from its input position by a specified amount.

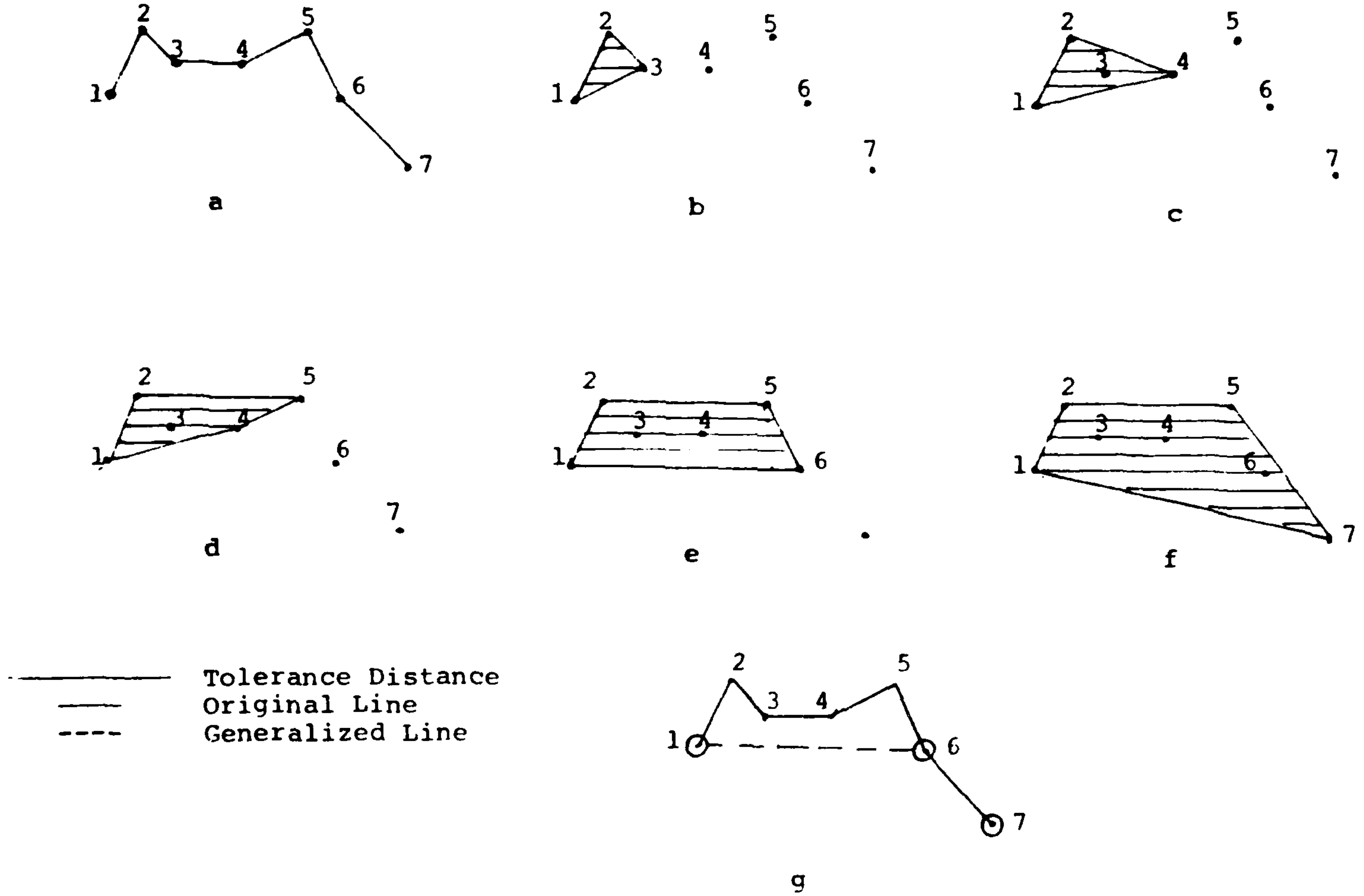


Figure 4.18. Dettori and Falcidiendo Algorithm. A) Original Line; B) to E) All Boundary Points Satisfy Tolerance for Some Side of Convex Hull, F) Distance for Point 5 to Side $\overline{17}$ Exceeds Tolerance. Therefore Point 6 is Selected. G) Generalized Line (1,6,7)

The earliest of these methods was suggested by Montanari (1970) who developed a mathematical programming approach to this problem. The idea here is that given the circular regions in which it is permissible to move points it is reasonable to demand that the generalized output be of minimal length. This leads to the following problem:

$$\min \sum_{i=1}^{N} \left((\hat{x}_{i+1} - \hat{x}_i)^2 + (\hat{y}_{i+1} - \hat{y}_i)^2 \right)^{1/2}$$

$$\text{s.t. } (\hat{x}_i - x_i)^2 + (\hat{y}_i - y_i)^2 < R^2$$

where x_i, y_i, $i=1,N$ are the original input vertices, $\hat{x}_i$, $\hat{y}_i$ are the vertices of the generalized curve, N is the number of vertices in the original data, and R is the acceptable circle radius. This is a problem with a unique solution which can be solved by a number of different programming techniques in relatively few iterations. Figure 4.19 shows a simple example of this approach. Methods for deleting output vertices which join straight line segments can also be specified.

In a recent paper Oommen and Kashyap (1983) discuss certain extensions to the Montanari algorithm. These include a preprocessing step to merge pairs of points for which the two regions of movement overlap and a method of choosing circle radii based on the total perimeter of closed curves.

Williams (1981) has developed an algorithm with the same sort of constraints but which do not use global optimization techniques. Williams' starting point was a version of Langs (1969) algorithm which he developed independently based on the work of Reumann and Witker and for which he presented an efficient implementation (1978).

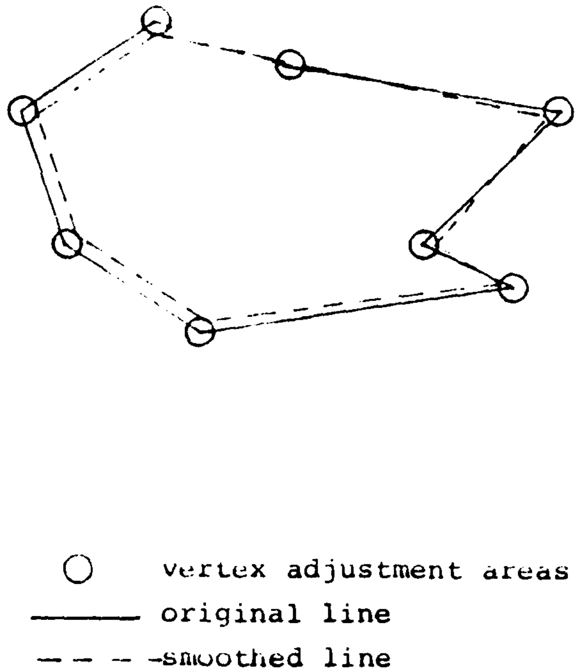


Figure 4.19. Minimum Length Smoothing

Williams notes that Lang's algorithm has a shortcoming in that it forces the approximation through curve points and consequently does not necessarily produce either maximum length vectors or optimally smoothed results. In his new paper, Williams varied his original implementation algorithm to free it from those point restrictions. The algorithm is driven by the goal of choosing the longest possible line segments for the curve generalization. The longest segments often are tangent to one of the point constraint circles. Thus the algorithm begins by finding for each curve point the set of line segments tangent to its constraint circle that also pass within the required distance of curve points on either side. As more and more points on either side are considered a non-increasing sequence of such sets is created. Unless only one line segment is necessary to approximate the original curve these sets will include a maximum length line segment at the step where considering one more point will produce a set which is null. When all such sets have been found for all points a complex search procedure chooses from them certain maximum length line segments for the final generalized line using techniques from dynamic programming. The result for one simple case is shown in Figure 4.20. A more complete description may be obtained from the original paper.

These algorithms can have an interesting interpretation in terms of the radius of the circles which are defined at each of the input curve vertices. Assume that this radius is equivalent to 1/2 the width of the symbolized line. Then the boundaries of the resulting symbolized line will either surround all input vertices or pass through them. This emulates the performance of a compiler who is creating a pull-up using a wide marker the width of which corresponds to the symbolized line weight required on the correctly reduced pull up. Under these conditions the compiler often controls marker movement to create a line which just picks up significant points on the input curve with the edge of the marker.

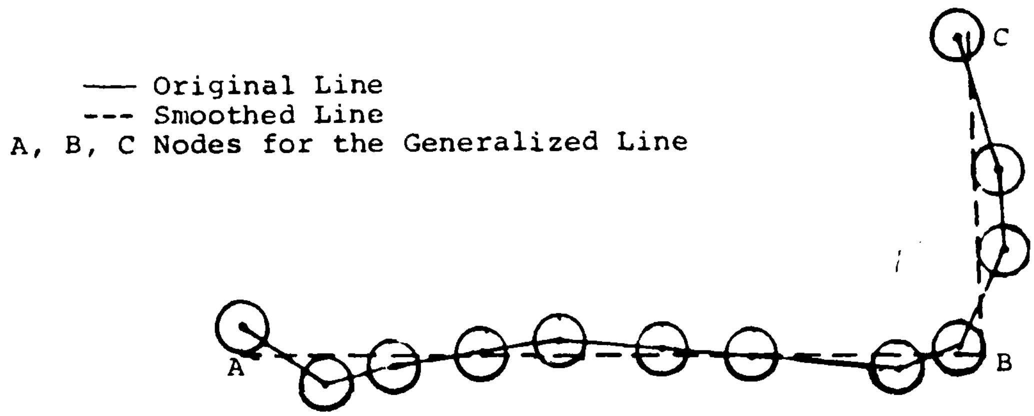


Figure 4.20. Williams Point Relaxation Algorithm

On the other hand, these algorithms obviously do not pick up the critical points exactly. Also, on relatively smooth curves they would have a bias to the convex side of any particular segment. This last tendency would be most pronounced in closed curves and might prove unacceptable.

4.7 DOMAIN TRANSFORMATION METHODS

The use of domain transformation techniques is mentioned much more often in the cartographic literature than it is actually described. In this approach the curve is decomposed into a linear combination of well defined, usually orthogonal, basis functions. The coefficients applied to the basis functions are determined by the shape of the original curve and the original curve may be reconstructed given these coefficients.

In Fourier transforms, a weighted summation of sine and cosine functions is used (Davis, 1973; Harbaugh and Merriam, 1973). Hopefully, the features which it is desired to delete in generalization may be identified in the coefficients applied to the sine and cosine functions. Some of these, usually corresponding to higher frequencies, can be eliminated before reconstructing the curve, thus obtaining a smoothed representation. Furthermore, information on the autocorrelation of the line is also contained in the transformed data and can be used to help determine parameters for other line generalization techniques (Robinson, et. al, 1978).

The only actual use of this approach in the cartographic literature is provided by Gottschalk (1971, 1973) who breaks the digitized input into x and y coordinates parameterized by arc length and applies a Fourier decomposition to both individually to obtain the width of the correct window to use in a moving average filter.

There are several problems with this approach. First, the analysis is global in scope, which assumes that the line character does not change fundamentally over its entire length. Second, the correlation between Fourier coefficients, line character and critical points is not well defined. Third, because of the cyclical nature of Fourier transforms these techniques can only be applied to closed curves. Finally, an approach which is applied to each component separately is going to produce different results depending on the coordinate system of the data. A simple rotation of the data will produce fundamentally different results.

Treating the x and y coordinates as inphase and quadrature components of a time varying signal as is done in radar and communications can deal with this last problem. In those fields two phase shifted samples of a received signal are combined into a complex number

$$x + i\ y.$$

The result is a sequence of complex numbers which may be decomposed using complex number representations of the Fourier basis functions. With this representation significant interpretations of the Fourier coefficients may be based on the magnitude of individual components $(x^2 + y^2)^{1/2}$ and the change in phase (ARCTAN(y/x)) between individual vertices. Many of these interpretations are independent of rotations of the data and with simple normalization techniques can be made independent of scale changes.

There are examples in the literature of such an approach used for shape recognition. (Cosgriff, R. L. 1960, Raudseps, J. G, 1965; Barrow and Popplestone, 1971; Moellering and Raysing, 1982; Tai and Chaing, 1982). Zahn and Roskies (1972) report upon a method which is based on a normalized cumulative angular function for a closed curve. The function measures cumulative angle changes as a function of the normalized arc length from an arbitrary starting point. The Fourier expansion of that function is then shown to contain information which is a function only of the shape of the curve and not of any combinations of translation, rotation, or change in size.

Sarvarayudu and Sethi (1973) use the same cumulative angular function but apply a Walsh series expansion rather than a Fourier series expansion. They report that this approach requires fewer components and has a computational advantage over the Fourier approach although it is sensitive to the choice of starting points.

These methods have been applied successfully for character recognition using closed contours. Whether they could produce acceptable results for line generalization is unknown. Such transformation techniques are usually not good at dealing with fine details. Mathematical sophistication is required to understand any domain transformation technique.

4.8 MATHEMATICAL FITTING

The fitting elements from a particular class of function or shape to a data set is a standard operation in many fields of engineering and social science. It is mentioned often in the line generalization literature but details of its use are hardly ever provided. Most of the results which are discussed in this section owe more to the fields of pattern recognition, image processing and computer graphics than to cartography.

These algorithms can be subdivided into two separate general classes: those that demand an exact fit to the input data points and those that require only an approximate fit. The former are useful in producing smoothed outputs from the segmented linear form characteristic of close views of digitized linear data. They are often used in image processing and CAD/CAM applications. They maintain all the detail in the input data.

The approximate fit methods have the capability of smoothing small details. They can be subdivided according to the method of fit and the type of function used in the fit.

4.8.1 Exact Fits

The best known method of obtaining a smoothed functional fit to data is the method of splines. In general a mathematical spline is a piecewise polynomial of degree K with continuity of derivatives of order K-1 at the common joints between segments of digitized data (Rogers and Adams 1976). Splines of order 3 with first derivatives matching at joints are most commonly encountered along with segments spanning only two points.

Another possibility is parabolic blending. Here four consecutive points are considered simultaneously. A smooth curve between the two interior points is generated by blending two overlapping parabolic segments. The first parabolic segement is defined by the first three points, and the last three points of the set of the four define the second parabolic segment. A blending function is needed to smoothly merge the two parabolic functions. Jancaitis and Junkins (1973) suggest one such blending function which guarantees first and second degree continuity at boundary points. They also suggest methods of enforcing certain derivative constraints at points along the input curve.

These methods are useful primarily in smoothing highly angular linear data. Unlike previous algorithms they produce more output points than input points. Since they do not reduce detail they will not be discussed further.

4.8.2 Approximate Fix Methods

Considerable effort has gone into developing methods to fit various mathematical forms to digitized data. The literature on this topic is extensive. Here we will be able to provide only an overview of the various methods which exist. Methods to be included are Bezier curves, arc fitting methods, and least square approximations.

4.8.2.1 Bezier Curves

A Bezier curve is a smooth curve associated with the "vertices" of a polygon which uniquely define the curve shape. Only the first and last vertices of the polygon actually lie on the curve; however, the other vertices define the derivatives, order, and shape of the curve (Rogers and Adams, 1976). The mathematical basis of the Bezier curve is a polynomial blending function which interpolates between the first and last vertices. The Bezier polynomial is based on a basic set of Bernstein polynomials of the form

$$J_{N,i}(t) = \binom{N}{i} t^i (1-t)^{N-i}$$

with N being the degree of the polynomial and i the particular vertex in the set of points. N vertices imply an Nth order polynomial of the form:

$$V(t) = \sum_{i=1}^{N} V_i J_{N,i}(t), \quad o \leq t \leq 1$$

where the V_i represent the curve vertices.

The key to understanding the use of these basis functions is the fact that the Bernstein function $J_{N,i}(t)$ has its largest value when $t=i/N$ and thus as t is varied between 0 and 1 each original input point dominates the shape of the curve over a certain small range of t.

Bezier functions were originally developed for use in interactive graphics, whereby a user could specify a curve by entering only a relatively small set of points. The computer could then efficiently produce a smooth shape near these points. This fact is indicated by the use of the term "guiding polygon" to describe the set of vertices which in line generalization problems would represent the input data. It is assumed that working interactively the user would be able to obtain the artistically correct results by varying the number of points originally entered. The results, for a fixed input, is a degree of line generalization, but like the spline functions discussed above, without a reduction in data set size.

4.8.2.2 Conic Form Fitting

There are a number of different algorithms for fitting parts of conics to digitized curves. Most of these were developed by non-cartographers.

Vanicek and Woolnough (1975) have suggested a method of line generalization based on fitting a function form called pseudo-hyperbolas to data. Their method, developed with plotter limitations in mind, uses a set of pseudo-hyperbolae functions of the form

$$y = \pm \frac{C1+C2}{X+C3}$$

After the first pseudo-hyperbolae is determined, the co- efficients for a new one are calculated so that it coincides with the end of the last selected line segment, and the axis of the pseudo-hyperbolae is orientated in the direction of the last line segment. Then the next points in the coordinate stream are examined until one falls outside a tube centered on the pseudo-hyperbolae and of predetermined width. Then a line segment is identified whose end point is at the intersection of the stream of coordinates with the pseudo-hyperbolae. The last point within the pseudo-hyperbolae provides the starting point for the next one. The whole process is initiated by an iterative process to find the first pseudo-hyperbolae which may be used to approximate the beginning of the curve.

Bookstein (1979) presents a method which works with a general conic of the form

$$f(x,y) = ax^2 + 2hxy + by^2 + 2ex + 2gy + c = 0$$

which includes circles, hyperbolae, elipses, parabolas and straight lines all as special cases. By use of certain normalizations of the conic coefficients he is able to obtain fits which match in first derivatives at "knots" where two different conics meet and which is invariant under translation, scaling, and rotation of the data. However, the segmentation of the input data is left to the user.

Pavlidis (1983) describes a more general method using conics in which automatic segmentation takes place. The segmentation involves hueristic rules for categorizing every vertex on an input curve as to whether it is a likely candidate for approximation by an interior part of a conic and whether a particular line segment is likely to be a place at which two conic approximations will join.

Not surprisingly the sides of vertices which are candidates for approximation by the interior part of a conic form angles closer to 180 degrees than to 0. Other rules specify that a vertex is classified as a break point if the ratio of its two sides exceeds some given threshold and that a side includes a break point if the vertex angles adjacent to it are on different sides of 180 degrees or differ by more than a given threshold. These rules are reminiscent of those presented in angle detection algorithms discussed in section 4.3. More information on these rules is available in Pavlidis' paper.

Once the vertices and segments have been categorized, the algorithm defines the various conic sections to be used in the approximation process. A very simple approximation to the distance from the conic section to the points approximated is used to determine whether a good fit has been obtained. In the case of a bad fit, an interval is subdivided and the two sections receive their own fits. Pavlidis does not attempt to obtain optimal fits. The most important reason from our point of view is the assertion that "it is very difficult (if not impossible) to devise mathematical criteria for approximation that agree with the human perception of high-quality approximation.... For applications where high qualitiy approximations are essential, it is necessary to include post editing of the results by a human observer."

None of the conic fitting methods discussed above has a direct relationship to the cartographic character of lines. Furthermore, they obtain their data compaction at the price of increased computation during plotting. They may have an important place in data compression techniques where it is desired to reduce the data set size and retain smoothness in the output curves at the same time.

4.8.2.3 Squared Error Fitting

Least square approximation methods are a classical technique of mathematics. Given a class of approximating functions of a single variable $F_k(x)$ and a set of observations $(x_i,$

y_i) the problem is to find a set of coefficients a_k which may be applied to the $F_k(x)$ function to create a combined function which minimizes the sum of the squared distance from the observations and the function. Usually the distance is measured parallel to the ordinate, but the distance may also be measured by the minimum distance from the points to the curve defined by the function.

When attempting to use this method for curve generalization the user is confronted with at least three major problems. The first is that a curve on a map, which may fold back on itself, is not likely to fit the standard functional model over its entire length.

The second problem is choice of approximating functions. The more complex and numerous they are, the easier it is to fit the data. Usually it is not desired to fit the data exactly so there needs to be a fundamental limit of the number of functions specified. To reach nontrivial decisions criteria must be used that impose a penalty that is an increasing function of the number of degrees of freedom of the approximating curve. A number of penalty functions have been discussed in the literature (Rissanen, 1978; Solomonoff, 1978). Pavlidis (1982) approaches this particular problem as one of pattern recognition and hypothesis testing. This problem has not been seriously addressed from a cartographic point of view.

A final problem is that it is almost certain that one fit cannot be used for the entire line. Thus the problem must be segmented in some manner, and the results for the various segments tied together. (This solves the first problems also.) In a practical approach this segmentation must be done automatically in order to free the user for other work. This is a difficult problem which is highly nonlinear when attempted optimally (Pavlidis, 1974).

Gottschalk (1971) presents a rather simplistic approach to least squares approximation. He suggests parameterizing the curve by arc length so that it is represented by the two functions x(t) and y(t)

This solves the first problem discussed above since both x and y are true single valued functions. However, the exact form of these two functions is not independent of the coordinate system used on the original map. For example, if the information on the map is rotated, then x and y will have significantly different forms which will affect the final results.

Gottschalk suggests the use of functions of the form:

$$x(t) = a_0 + a_1 \sin t + a_2 \sin 2t + a \sin_3 3t + a_4 \sin 4t$$
$$+ b_1 \cos t + b_2 \cos 2t + b_3 \cos 3t + b_4 \cos 4t$$

and also of the form:

$$x(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + \ldots$$

with y defined correspondingly. He divides the input data rather arbitrarily into segments 20 points long and specifies constraints to guarantee that the approximating functions agree in value and first derivative at joints.

Jancaitis and Junkins briefly address this problem. They reject the decomposition approach due to the complexity of the problem it creates and difficulties in the interpretation of arc length once the fit has been calculated.

Pavlidis (1977) considers automated segmentation techniques in great detail. Starting with the goal of fitting straight lines or more complex forms each of which satisfy some maximum least squares error constraint, he investigates a number of ways of breaking the curve into suitable partitions. Two fundamental methods are described. In merging schemes the original curve is searched for the longest segment which may be fitted at one time with a sufficiently small error. As the process is repeated, the entire curve is approximated. In splitting or subdividing methods, large arcs are subdivided until a sufficiently small error of approximation is achieved. Usually these latter schemes work on the principal of bisection. Segments are divided in two as long as the required error constraint is not satisfied.

These two approaches to this problem are similar to the two methods of finding dominant points and associated segments which are described in Section 4.3. In fact, these methods usually create knots where two segments meet at vertices which would be classified as dominant in that section. The opposite case is not true, since often knots are established at points of low curvature. This may be detected by measuring the sensitivity of the error terms to changes in knot positions. Near low curvature positions this change will be small.

The two methods may be combined. Pavlidis and Horowitz (1973) describe a split and merge algorithm which solves the following problem: Given a set of points $S = (x_i, y_i)$ determine the minimum number n, such that S is divided into n subsets S1, S2, etc. where each of the data points are approximated by a straight line with an error norm less than a prespecified quantity E_{max}.

4.8.3 Mathematical Fitting Algorithm Evaluation

Exact fit methods and Bezier curve techniques serve a different purpose than the other algorithms discussed in this report. They are most appropriate in interactive graphic operations. They may have a useful place in an automated cartographic system but will probably not be utilized in line generalization.

The algorithms discussed under approximate fit methods are more appropriate for generalization operations. The mathematical forms used by these algorithms might also be useful for feature detection operations. However, the techniques investigated for this report have not developed to the degree necessary for use in cartography. Much research would be required to define the correct algorithm implementations for use in automated line generalization. Even with the best implementations operator control is likely to be uncertain.

4.9 EPSILON FILTERING

An algorithm which belongs in a category of its own is epsilon or epsilon circle filtering. This method was originally developed by Perkal (1966) during the 1950's as a tool for the measurement of length of empirical lines. Later he suggested using it as an objective method of generalization.

When used in generalization the algorithm involves rolling a circle of radius epsilon along both sides of a curve. The path of the edge of the circle defines a generalized curve which is dependent on which side of the original curve the circle is rolled. In complex regions a residual zone will be left between the two sides. An artist's concept of this procedure is shown in Figure 4.21.

The residual zone between the two generalizations provided by rolling a circle on different sides of a curve is a fundamental problem with this approach. Since it is easy to create a curve for which this region may be arbitrarily large, this problem is not something which may be ignored in practice. Furthermore, the task of "rolling a circle along a curve" is not easily done by a computer. Thus any direct utilization of this approach in line generalization would seem to be very difficult.

Recently Chrisman (1983) suggested a method of generalizing which is inspired by Perkal's work. In Chrisman's

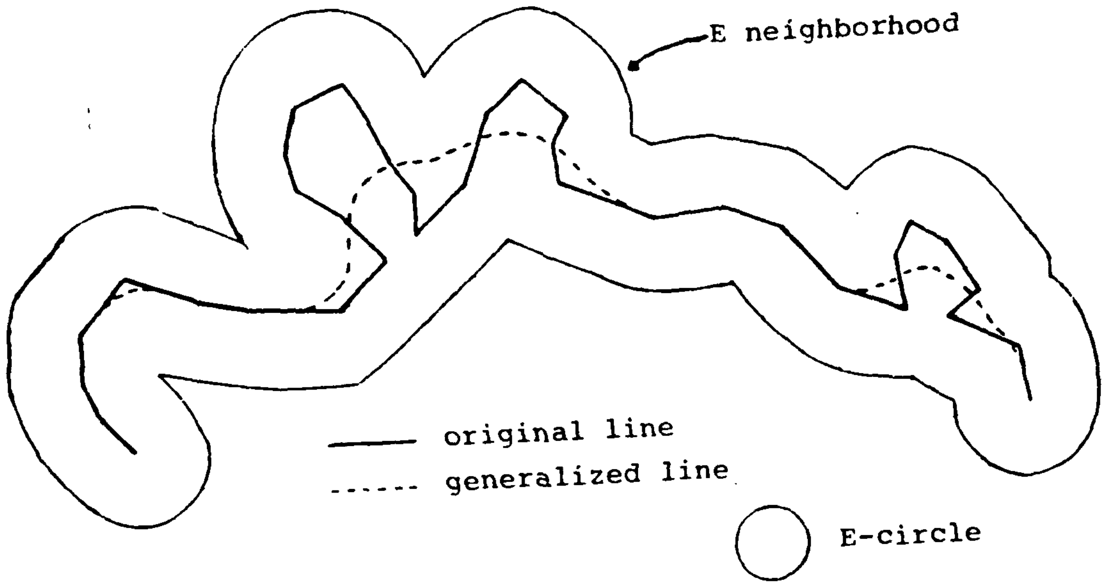


Figure 4.21. Perkal Epsilon Circle Algorithm

algorithm, epsilon circles become clusters of points within epsilon of each other. These clusters are thinned until no point is left within epsilon of any other point, a process which requires the movement of some points.

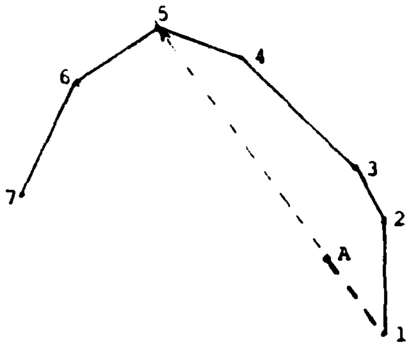
The exact processes involved in these clustering and thinning operations are not fully described. They are implemented in a software package named WHIRLPOOL (Dougenik, 1980) which is part of the ODYSSEY system at Harvard University. This software is proprietary and cannot be examined without purchase.

The example provided in Chrisman's paper involved the generalization of polygonal boundaries which reflects the fact that the WHIRLPOOL algorithm is oriented towards areal data. Thus, there are no examples of this algorithm being utilized for normal line generalization tasks. It does not seem that either Perkal's original ideas or Chrisman's adaptions are likely candidates for general purpose line generalization software.

4.10 OTHER METHODS

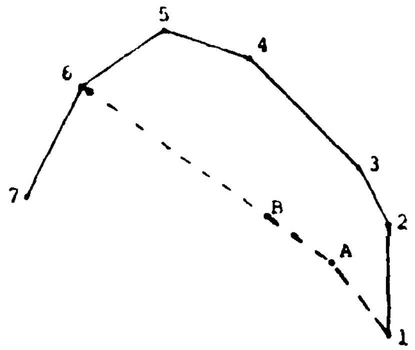
The methods discussed in this section do not fall into any commonly defined methods but are well known. Boyle (1970) develops a "forward look" interpolation method where the line is "aimed" at a section of line n points ahead. The movement is 1/nth of the distance from the last selected point to the point n points ahead. The anchor then moves to the selected point and the procedure is repeated. This is shown in Figure 4.22.

A method developed by Brophy bases generalization on circles inscribed in convex segments of the input curve (1972). This is somewhat similar to Perkal's approach described in the last section. First a triangle is created with a primary vertex at the point being considered for movement and with its other two vertices corresponding to the n-th point preceding and following



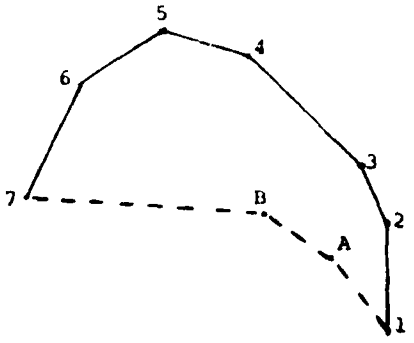
STEP 1.

N=4
A line is aimed from point 1 to point 5, 4 points ahead. Point A is chosen such that $\overline{A1} = (\frac{1}{4})\ \overline{15}$



STEP 2.

Similarly Point B is chosen so that $\overline{AB}$ is one-fourth of $\overline{A6}$



STEP 3.

Generalization includes end point.

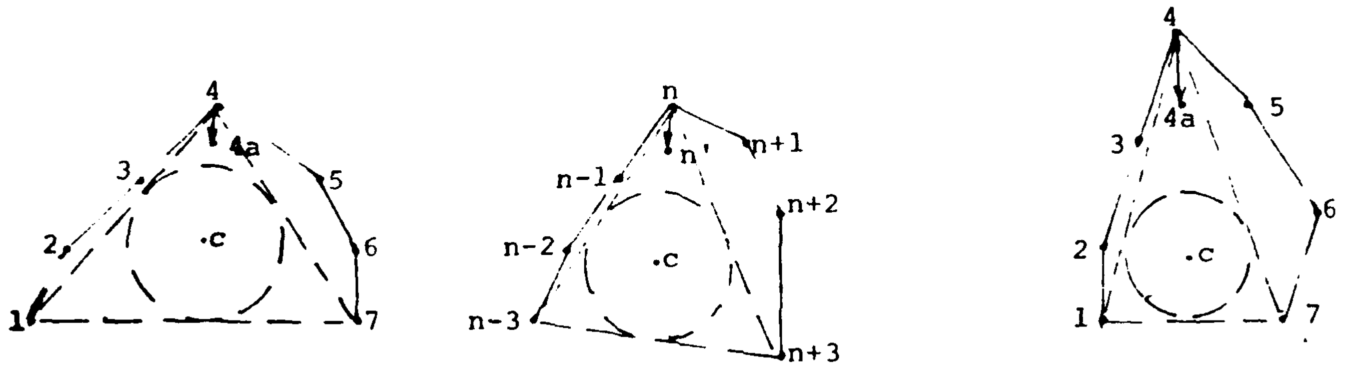
——— Original Line
----- Generalized Line

Figure 4.22. Boyle's Forward Look Algorithm. N=4

the point under study. A circle is inscribed in that triangle which will vary in size with the size of the angle at the primary vertex. The center of the circle will move away from the primary vertex by an amount directly dependent upon the size of its angle. This is indicated in Figure 4.23. Generalization involves moving the primary vertex towards the center of the circle by an amount equal to 5/6 of the distance to the center of the circle. The amount of generalization for a curve is varied by changing the value of n used in picking triangle vertices.

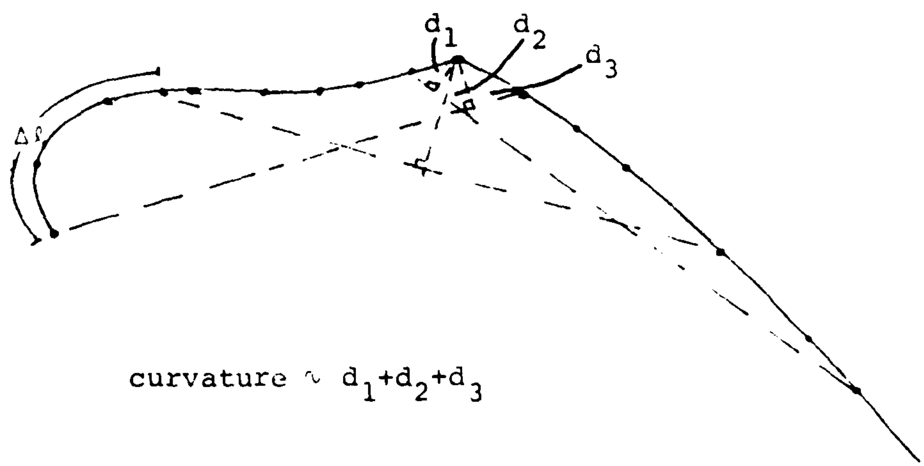
Brophy's algorithm is motivated by an attempt to easily approximate the curvature at a given point. Johanssen (1973) develops a slightly different method with the same goal in mind. He proceeds as follows: a chord of fixed length l is stepped along the curve, moving at each step a predetermined distance l. Movement is such that for each vertex on the input curve the perpendicular distance from successive chords can be calculated and summed. This sum is considered to be proportional to local curvature at a point. Now, if the goal is to emphasize points of high curvature, points of small local curvature are detected and removed from the input set. If the goal is to smooth the whole curve, then points of high local curvature are detected and removed. This is shown in Figure 4-24.

One other method of generalization has received considerable attention in the last few years. This is the use of fractional dimensionality or fractals, first explored by Mandelbrot (1977). In a break with standard mathematics, Manderbolt treats dimension as a continuum in which the integer Euclidean dimensions represent limiting cases. For example, digitized curves in the plane are assigned fractal dimension greater than or equal to 1 and less than 2. This has presented a new and often useful way of studying widely disperate phenomena such as the utilization of computer chace memory, Browning motion, and the characterization of cartographic features such as coastlines.



——— Original Line
c Center of Inscribed Circle
n' or 4a New Position of 4 or N (5/6 of $\overline{nc}$)

Figure 4.23. Brophy's Circle Algorithm Showing Decreased Arch Size with Higher Point Curvature



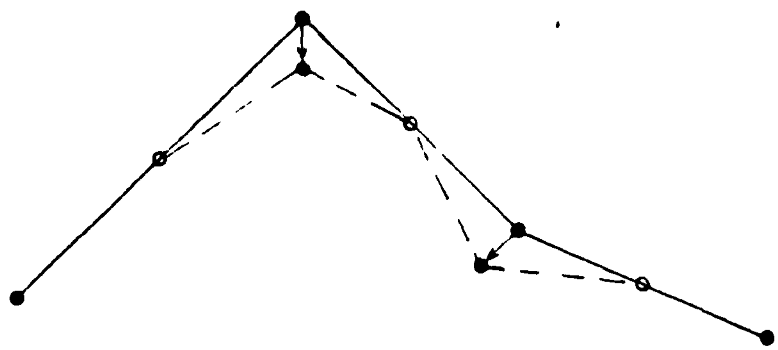
curvature $\sim d_1+d_2+d_3$

Figure 4.24. Johanssen's Local Curvature Calculation Algorithm

An important concept in the study of fractals is self-similarity. In its most pure form self similarity implies that any piece of a curve or figure can produce an exact replica of the whole. Mandebrot demonstrates certain real world examples in which this is true at least over a range of refinements. However, it is not generally true of geographic features over all scales (Goodchild, 1980).

Dutton (1981) has developed an iterative algorithm which produces fractal models of existing lines. According to Dutton this algorithm permits exaggeration of features and the introduction of small scale features as well as the elimination of features during the generalization process. The process involves both the introduction of new points in the curve and the standardizing of all angles along the curve. Four parameters are required to control the operation. One defines the desired standard angle and a second determines the degree to which all angles are equivalent. Two other parameters restrict operation of the algorithm over parts of the curve made up of particularly long or short segments. The basic operation involves computing the midpoints for two adjacent segments, connecting these to form a triangle with the common endpoints at the apex, and then moving the apex vertex an amount determined by the first two parameters and the shape of the triangle. This is shown in Figure 4.25 where two vertices are moved, one to decrease an angle and the other to increase an agle. The segment midpoints now become added vertices of the digitized curve and are themselves subject to angle changes in subsequent adjustment steps.

The use of fractal measurement provides an interesting approach in characterizing cartographic features. Still, while preserving general line character, Dutton's algorithm is not intended to provide geographically accurate representations. This is a significant problem that makes it difficult to image using



● Input vertices in original and generalized positions

o Segment midpoints and new vertices

——— Original line

— — — Generalized line

Figure 4.25. Dutton's Fractalizing Algorithm Showing Adjustment of Two Vertices

this technique for topographic map production. Dutton suggests this technique may be more useful for thematic mapping than for other cartographic applications. Furthermore, the need for four parameters makes this technique potentially difficult to control.

5.0 FEATURE DISPLACEMENT ALGORITHMS

The literature on feature displacement algorithms is much sparcer than that for line generalization. There are at least three reasons for this. First, the line generalization problem is conceptually easier. Second, it is relatively easy to implement and test line generalization algorithms. Third, there are many results in disciplines such as image processing, which may be applied to cartographic line generalization and which help to motivate work in that area, but there is no directly related work in other fields for the feature displacement problem. The only field in which significant research work is being performed which deals with a similar problem is the automated layout of printed circuit board and integrated circuits for electronic applications.

There is, apparently only one complete algorithm in the cartographic literature for automated feature displacement involving complex features and arbitrary symbolization. Even that algorithm does not deal with area features and certain other details required by a full implementation.

There are other papers which deal with parts of the problem such as name placement on point and line features. These are discussed in the following sections. There are also a number of papers which discuss interactive displacement. These are referenced in the bibliography but not discussed in this report.

5.1 LICHTNER'S ALGORITHM

Lichtner (1978) presents an algorithm for dealing with point or area features with respect to enlarged linear features. The method he develops takes advantage of the small scale of the output map to hide minor distortions created during the displacement operations.

The first step is to record the largest displacement required due to changes in symbolization of linear feature. This is indicated by V_o in Figure 5.1 and is just the change in symbol size of the linear feature measured with respect to the center line of the feature. Displaceable features immediately adjacent to the linear feature might need to move that much. Other features farther away are allowed to move smaller distances until some maximum range T is reached at which no displacement effect results. The maximum range is chosen to keep distortion caused by the movement method limited.

The displacements V_i at the points of the feature P_i falling inside the displacement area VZ decrease linearly from the maximum amount V_o to zero. From Figure 5.1 it is possible to obtain the equation for V_i

$$V_i = V_o \left(1 - \frac{S_i - \frac{B_a}{2}}{T} \right)$$

As it is not the center point of feature, but all points digitally recorded (e.g. corners of buildings) which are displaced, a constant distortion of all distances at right angles to the axis PA of the primary linear feature takes place.

A depth of displacement zone of $T=11*V_o$ is based on empirical research. This constrains maximum distortion to less than 10%. These distortions are minor at the scale of the output map under discussion. Other ranges might be appropriate for other maps.

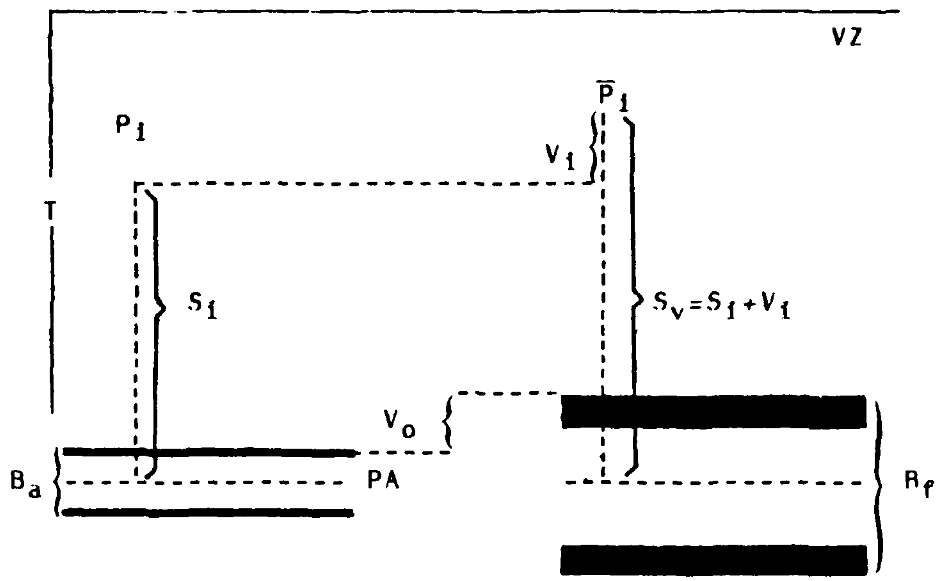


Figure 5.1. Lichtner's Algorithm

Exactly how this program would deal with multiple linear features affecting the same feature is not clear. First do one and then do the other? Or could there be a priority for the various linear features?

This algorithm also does not deal with other types of displacement problems such as two linear features conflicting directly with each other or with the situation in which point or area features moving away from different linear features come into conflict with each other.

5.2 KRISTOFFERSEN'S ALGORITHM

Kristoffersen (1980) discusses a method of displacement designed to work on a low density map containing only points. A given type of symbol represents a given object category.

For the cases described, a map sheet contains from 20 to 50 symbols and 70-85% of the conflicts which occur involve less than 5 symbols. Manual means are provided for resolving more complex problems.

As objects are first encountered by this algorithm, a table of object coordinates is created. This table is sorted on one coordinate. Groups of close symbols are now easily found by checking the second coordinates due to the fact that all symbols are roughly the same square size. Members of a conflicting group are linked together. Each group is then processed separately. There are 4 displacement directions: East-West-South-North corresponding to the maximum number of symbols that may be handled automatically in one group.

Displacement distances are always equal to the size of one symbol. The first step in finding displacement directions is to create the smallest possible rectangle around the group of symbols. If a feature touches only one side of the rectangle and that side is touched only by that feature, displacement is chosen to be in that direction. This is indicated by features 1 and 2 in Figure 5.2.

Features touching corners of the rectangle can move in either of two directions as indicated by feature 3 in Figure 5.2. A decision is based on movement directions assigned to other features which touch only one boundary or which are interior in the rectangle.

Symbols not on the boundary of the enclosing rectangle (there may be two of them) are referenced to rectangle bisectors in the North-South and East-West directions. Displacement directions are chosen so as not to conflict with those asigned to border symbols and so as not to cross the rectangle bisectors. (There is room for ambiguity still, apparently arbitrary decisions are made if necessary.)

This is a very simple program which apparently completely resolves a straightforward problem. One can imagine extensions which would deal with larger group sizes. One approach which comes to mind is defining a more complex enclosing polygon, a hexagon for example.

Note that maps which may be divided into small separate problem areas are not easy to find in real world. Of course the same size of the input data set and the uniform symbols sizes and hierarchy makes this a straightforward problem.

5.3 NAME PLACEMENT ALGORITHMS

A number of algorithms have been developed to handle a small subset of the problem: automated names placement and conflict avoidance. An algorithm by Hirsch (1982) represents the current state of the art in the area of names placement around point symbols.

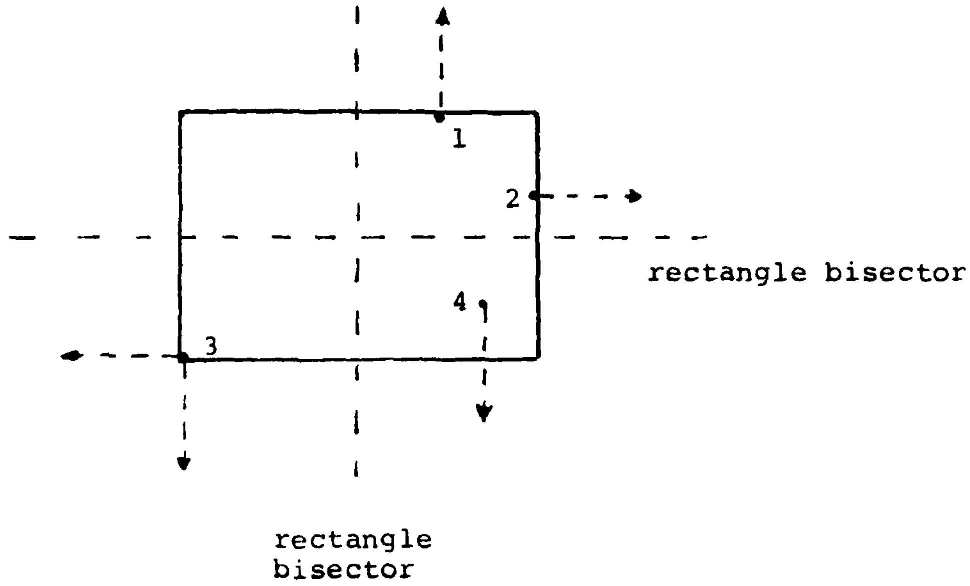


Figure 5.2. Kristoffersen's Algorithm

His algorithm is designed to satisfy two objectives: 1) to place names so that they do not overlap; and 2) to place names so that each clearly refers to its point symbol.

Hirsch suggests an iterative approach based on vector driven movement of the name and a circle defined around each point with which to reference the name position as shown in Figure 5.3. Movement vectors are constrained so as to move the names placement around the circle which has radius equal to the letter height. Certain positions around the circle are specified as superior for names placement and are given preference in this algorithm (Imhof, 1982).

Processing includes sorting of circles and the rectangles defining names size, overlap detection algorithms, and the iterative movement calculations. At each pass movement vectors are defined for each name which define quantities for movement away from conflicts. If a name is found to be in conflict with multiple other names and symbols, the movement vectors required by each conflict are summed to create a total movement vector. These are not used directly since the names must be tied to the point features through the surrounding circles, but as guidelines for small movement and placement around the cirle according to predefined rules. These rules include special conditions depending on whether the current name position and the movement vector are in the same quadrant defined by the point center and whether the name is in certain special zones around the circle. (Option exists for large movement also if small incremental movements do not result in resolution after a certain number of iterations.)

There are several interesting points about this algorithm. First, it attempts a global aggregation of information by developing a movement vector for each name simultaneously by

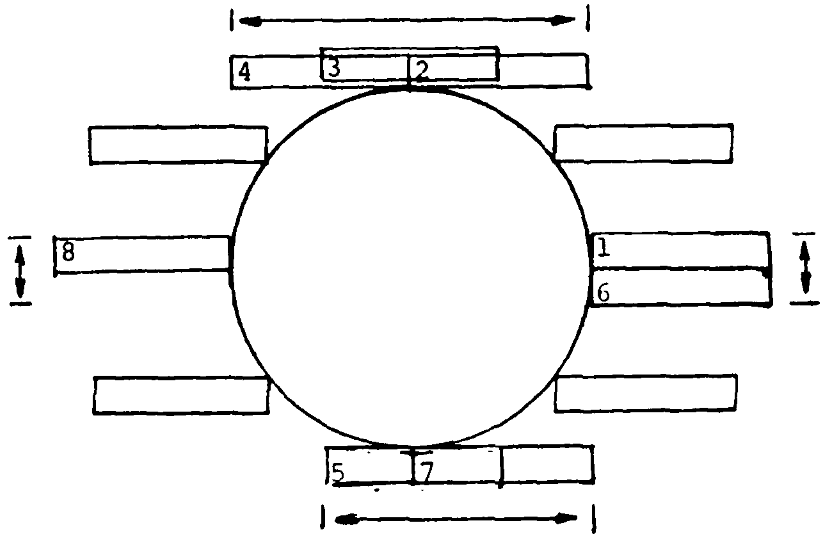


Figure 5.3. Preferred (Numbered) and Intermediate Name Positions on the Circle Showing Movement Within Special Zones (after Hirsch, 1982)

means of vector addition. In some ways this reflects the aggregation of information which is performed by a human compiler. Also, in its vectorized approach it is similar to the approach used by Christ for general displacement.

This algorithm is defined only for point data. Apparently in practice it may not converge to a final solution using the first iterative method. Hirsch suggests a switch to method two at that time. However, this is apparently not automatic in the current algorithm.

5.4 CHRIST'S ALGORITHM

Christ (1978) has developed a rather sophisticated program for feature displacement. It handles arbitrary symbolization specifications for point and line features.

The algorithm is based on a word map of the entire map. For the implementation described this is of size 1024 by 1024. Each word corresponds to a location (actually a tiny area) on the map. For each feature on the input data the algorithm locates the words in the word map which correspond to the feature's (true) input scale location. Certain bits within these words are set to record that the word corresponds to a feature symbol and others to record the feature priority.

For point features a circle is constructed around the center of the feature of diameter:

$$b + R(z-b)$$

where: b is the size of the old symbol and z is the size of the new symbol and R is defined below. The area of the circle corresponds to a "free space" for the feature. The algorithm is driven by the requirement that this space be reduced no more than a certain factor. Christ divides features into 3 separate classifications based on size for which the reduction factor varies from 100% to 0%. A reduction factor of 80% (meaning that the free space can be reduced to 80% of its prior extent in a particular direction) implies R = 5 (R=1/(1-.8)). Within this circular region extending away from the center of the feature each word is encoded with the feature priority and with a displacement vector, the magnitude of which decreases in a linear manner with distance from the center; and the direction of which is oriented in one of 16 different rays (every 22.5 degrees).

The net effect of this is to mark a circular region as used. The central part of the circle corresponds to the actual symbol and the annular region about that corresponding to the "free space" which will be reduced by 80% (1-1/5) by the resymbolization.

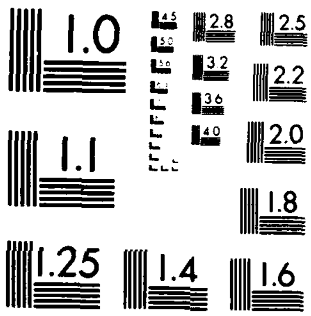
Perhaps more illuminating is the realization of what is not stored. Nowhere is any effort made to store the symbol or even its type (line or point). The only thing that is stored is information indicating which way you would have to move to get far enough away from another symbol to keep "free space" above 80%. A cross section through a feature showing the displacement effects measured vertically is shown in Figure 5.4.

Line features are handled with a little more subtle approach. Essentially the same procedure is followed but instead of using a circle (as one would about a point feature) a band is used. The band has the same width as the above circle would have diameter. The computational method used to locate these points is described in the paper using this idea of an "outrigger". The author states that the outrigger is perpendicular to a given line segment at only one point. It seems that in his implementation the outrigger changes direction in some continuous manner.

AD-A143 250 MANUAL AND AUTOMATED LINE GENERALIZATION AND FEATURE DISPLACEMENT(U) ZYCOR INC AUSTIN TX S ZORASTER ET AL. JUN 84 101-43-16 ETL-0359 DAAK70-82-C-0149

2/2

UNCLASSIFIED F/G 5/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

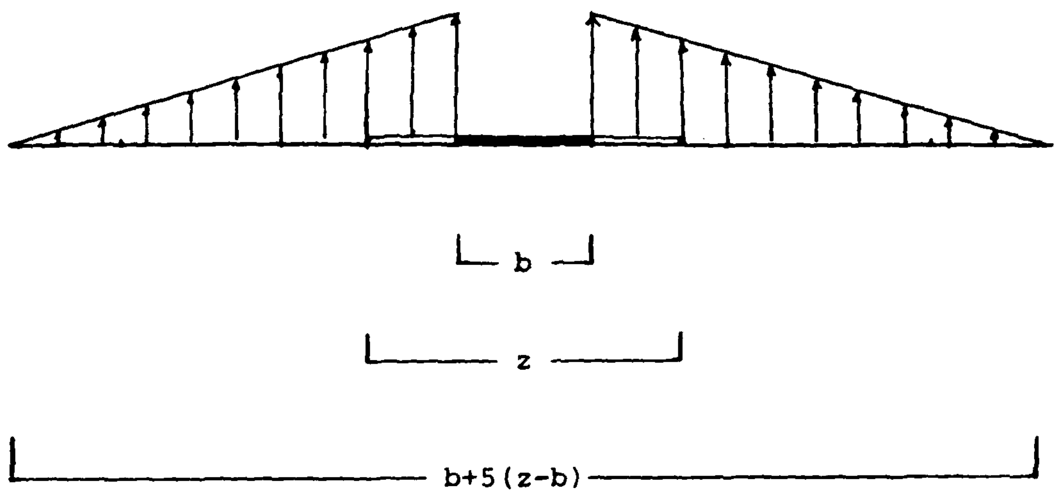


Figure 5.4. Displacement Effects Represent Vertically for Christ's Algorithm

As the circular regions or bands around linear features are defined for each feature it is possible that it will be necessary to mark as used a word which already has been used. This implies that the free space of two symbols are in conflict. When this happens a second bit is set and the vectorial sum of the current "displacement" vector with the vector already present in this pixel is calculated and stored.

When the affected areas have been processed and marked it is possible to locate areas of potential conflict between features, by the special set bit, and process them. The information in these pixels is somewhat analagous to a vector field defined in physics with the information at each pixel showing the strengths of the displacement efforts required. Those with no overlay require no displacement. Those with an overlay require displacement.

Although the paper is unclear on this point, the displacement effect is apparently proportional to the magnitude of the added displacement effects at each overlayed pixel. By means of a digital contouring algorithm it is possible to define directions of steepest displacement. These are then (apparently) smoothed out by adjusting the displacement effects on the pixels with a lower priority back towards the center of the feature. The adjustment step is such that even the clear regions in the center of the feature area are now given displacement effects so as to indicate movement for that feature away from the feature(s) with which it would be in conflict.

Now the features are added to the map. Long, linear line features are broken into shorter segments (by interpolation) and placement is begun. In areas where there is no conflict, the symbol may be directly placed onto the map. In areas where a conflict has been found, the symbol with the highest priority is placed in its correct location. Subsequently, features of lower priority are displaced by the amount specified by the displacement vector at that point.

There is a large amount of information about this algorithm not fully explained by the paper. This is not an easy algorithm to describe and the fact that it has been translated from German adds to the difficulty. Among the simple problems is that no mention is made as to what to do when the free spaces of two features of equal priority overlap. Presumably the adjustment of the displacement vectors is applied to both equally. Also, no discussion is provided of cases where the displacement of two features which originally did not conflict brings them into conflict, or for that matter of a feature of low priority trapped between two features which are moving toward it from opposite directions. A further problem with this paper is that no discussion is given of how to work with area type features such as large airports or city symbols.

Most importantly, there is no good explanation of how the algorithm carries out the adjustment of the displacement effects in overlay areas. The discussions about overlay contours is not very helpful. One way to think about this process is as a map filtering problem such as filtering of a DEM in which the displacement effects are gradually moved away from areas of overlap.

5.5 CIRCUIT DESIGN

Outside of cartography, there is a related problem involving two dimensional object placement for which much progress has been made. This involves the automated placement of circuitry packages on printed circuit boards or integrated circuits (Hanan and Kurtzberg, 1972). In this discipline the problem is one of placing rectangular objects onto a two dimensional board in such a way as to minimize or maximize certain functions related either to their interconnections or to their density. Among the objective functions considered are the minimization of the total amount of wire required to interconnect the packages; the minimization of wire crosssings required to interconnect the packages; and the maximization of the number of packages which may be placed on a board.

Solution techniques for this class of problem have tended to be both iterative and hueristic. One general approach involves placement of all packages on the board and then iteratively adjusting the configuration by switching pairs which may improve the value of the objective function. Another approach involves placing packages with the most interconnections to other packages first and then placing less connected packages in the remaining space. Implemented algorithms usually involve a combination of different approaches. Some of the iterative methods involve vector directed movements which are reminiscent of the techniques used in the algorithms by Christ and Hirsch.

The problem just described is considerably different from the cartographic feature placement problem. In many of the differences the cartographic problem is more difficult. Not all cartographic features are rectangular nor can they be expected to be orientated parallel to the axis of a rectangular grid which may be assumed in package placement algorithms. Cartographic objects have fewer degrees of freedom in their movement. Finally there is no direct correspondence between the objective functions required in the two fields.

There are ways in which the cartographic problem is easier. Density of features over the whole map would tend to be less than that encountered in board or chip layout. And a starting placement of objects near to their true geographic location can provide an initial configuration which is physically close to a final solution. The most important point in considering this related problem is that the two seem to be of roughly the same order of complexity.

6.0 APPROACHES TO ALGORITHM EVALUATION

Ever since the inception of automated cartography, cartographers have been developing algorithms to perform line generalization and feature displacement. The variety of algorithms specifically available for line generalization is formidable, however the relative quality of these algorithms has not been studied in any great detail. Part of the reason for this failure is the difficulty in measuring quality in such a subjective field as map compilation.

Any method used to perform generalization tasks must provide visually satisfactory results, the exact meaning of which is extremely difficult to define. However, it does not follow that it is impossible to provide a framework in which to judge algorithm performance. This section describes an abstract set of cartographic and computational or algorithmic constraints which "good" line generalization and feature displacement algorithms might satisfy. Additionally, a discussion of research guidance in this area is provided.

6.1 CARTOGRAPHIC CRITERIA

Algorithms which are developed for line generalization and feature displacement can satisfy many different measures of cartographic quality. Some of those to consider include:

- preservation of map character and accuracy
- a focus on global features
- the ability to vary the amount of generalization as a function of feature type
- the ability to perform exaggeration and relocation of map elements and features.

6.1.1 Preservation of Map Character and Accuracy

The locations of features should not change drastically during generalization or displacement. Strict DMA specifications must be satisfied by an automated compilation system. However, simplistic quantitative limitations are not the answer to the whole problem.

The generalization process must be done with proper attention to the necessity of preserving the significant character of the map features. For displacement this can mean that groups of features which form a recognizable unit should be moved together even if not required by existing standards. It also means that coalescing of features and the feature selection process may be significant parts of any displacement algorithms.

For linear features there has been much recent interest among cartographers in determining those points on a line which are most significant in characterizing the line. This work has provided an extension of previous work in perception and cognition (Attneave, 1954; Dent, 1972; Freeman, 1978). This has led to the conclusion that there are characteristic or critical points which are perceived with a high degree of repeatability by both cartographers and non-cartographers. These points contain the most significant information regarding the nature of the line and thus should be retained in the generalization process (Jenks, 1980; Marino, 1978, 1979, White, 1983).

It is unlikely that focusing on critical points for a single line fully defines the correct way to study this problem even though the vast majority of line generalization algorithms focus on one line at a time. It is the features represented by contours, for example, which are important rather than the contours themselves (Imhof, 1982).

6.1.2 Global Operator

Computer-assisted algorithms for line generalization are most often applied to lines rather than features. It is easier to generalize a single line than to generalize a feature containing that contour line. Unfortunately it is the features rather than the lines which are significant.

The association of features is especially significant for displacement. A proposed solution must not only identify and resolve a conflict, but identify and resolve any secondary conflicts resulting from the solution of the initial conflict.

Thus a good algorithm should be in some sense a "global operator", able to continuously aggregate information from surrounding geographic features in the generalization process. Unfortunately, algorithms which aggregate spatial information are extremely hard to create. This is a process which humans perform quickly and easily and which so far has been performed by computers only with great difficulty. Robust techniques to identify common cartographic features do not exist. Problems with developing such algorithms in well researched related fields as image processing and electrocardiogram analysis point up the difficulties in this work.

6.1.3 Able to Vary the Amounts of Generalization

Consideration must be given to the variety of different features to which an algorithm may be applied. These can include contours, drains, road networks, political boundaries, bottom contours, etc. Each may require a different amount of generalization on the same product. Furthermore, the same type of feature may require differing amounts of generalization depending on what parts of the map the feature appears.

6.1.4 Exaggeration and Relocation

Generalization may be assumed to be merely the reduction of existing feature data, but this understates the problem in practice. Often the goal is to develop a representative pattern at the same or a reduced scale and this may involve both exaggeration and relocation. For example, a small spit of land containing an important feature might be exaggerated so that the spit and feature could be retained on the smaller scale map. Additionally, a stream that crosses a road a number of times in a short distance would be relocated to the side on which it predominantly resides. Both exaggeration and relocation may be necessary for developing a complete automated system.

6.2 ALGORITHMIC CRITERIA

Algorithms which are developed for line generalization and feature displacement can satisfy many different measures of algorithmic quality. Some of those to consider include:

- predictable reduction in data
- invariant with respect to mathematical operations
- predictably controlled by simple parameters
- modular to meet different map specifications
- computationally fast

6.2.1 Predictable Reduction in Data

The primary goal of any algorithm must be to improve visual representation. However, a map produced from sources requiring a factor of 5 scale change implies 25 input maps to cover the same area. An algorithm which does not significantly reduce the number of data points during generalization will produce digital products which may strain the storage capabilities of the associated computer system.

An algorithm which could reduce the number of data points in a predictable manner would ease the problem of specifying required data storage requirements of an automated environment.

6.2.2 Invariance With Respect to Mathematical Operations

Algorithm invariance with respect to data manipulations which should not affect the results is a well recognized optimality requirement in many disciplines. For example, most well known statistical procedures such as T-tests and F-tests are invariant with respect to scale changes in the data.

Similarly we expect a line generalization procedure to be invariant with respect to many common manipulations of the data. These include choice of starting points, rotations of the data, and choice of units for data representation. A number of existing algorithms fail to satisfy this requirement. For example, algorithms which perform complex independent manipulations on the x and y coordinates of the digitized contour usually are not invariant.

In the same way we would expect displacement algorithms to produce similar results independent of the order in which data is provided.

We note however that a human cartographer may not create exactly the same results given data which has been rotated by 90 degrees. Thus absolute invariance may not be needed. Furthermore, absolute scale invariance is not desirable within a given map. The generalization of a curve is likely to be strongly correlated with the size of the display at target scale. For example a small spike on a short insignificant line may be completely eliminated while a large spike of the same relative proportions on a longer more important line may need to be retained.

6.2.3 Predictable Control

None of the algorithms discussed in chapters 4 or 5 can be used without care in the generalization process. Human direction will always be required. Furthermore, in a large scale cartograph production environment, it is to be expected that compilers may not be fully experienced cartographers or be familiar with all applications. Thus good algorithms must have control parameters which are 1) understandable to non-expert users and 2) easy to correlate with the amount of detail in the input data and the cartographic goals of the compiler.

Some very simple algorithms are actually difficult to use. This is because their very simplicity makes their performance difficult to correlate with cartographic goals.

6.2.4 Modular to Meet Different Map Specifications and Data

Algorithms supplied to DMA must satisfy the demands created by a multitude of specifications and data types. Algorithm developers will not be able to foresee all possible combinations of demand or types of problems which will arise. Therefore algorithms must be modular in approach so that they may be performed in varied order and combined arbitrarily to meet compilation requirements. A choice of algorithms for a single task will allow cartographic license to continue to play a significant part in map compilation.

6.2.5 Computationally Fast

The size of the task DMA envisions performing with automated techniques makes speed very important. An algorithm which performs "perfect" line generalization and feature displacement will be unacceptable if it requires excessive computer resources.

It is commonly assumed that computer resources will be become cheaper in the future; but at the same time, DMA use of computers will become greater. It would be dangerous to rely only on hardware improvements to make a slow algorithm acceptable at some hypothetical hardware milestone in the future.

A distinction should be made between interactive and non-interactive generalization operations. If algorithms are developed which have a high degree of reliability then performing generalization operations in a batch submission can take large amounts of computer time without undo burden on the compiler.

6.3 OTHER SOURCES OF GUIDANCE

Although much has been written about these problems in the cartographic literature authors seldom provide hard rules. Words such as critical points and feature character are presented without being defined. (Defining them rigorously may be impossible.) Algorithms are developed which are intuitively reasonable but which are never subject to rigorous testing.

It is only within the last 2 or 3 years that serious consideration has been given to comparison of line generalization algorithms (Jenks, 1980; Marino, 1978, 1979; White, 1983; McMaster, 1983a, 1983b). Part of the reason may be that obtaining a basis against which to compare algorithms involves a tedious survey of individual manual generalization techniques. Another possibility is that in practice people care less about the quality of these algorithms than they do in theory.

Marino carried out the first research in this area by asking both cartographers and non-cartographers to pick "critical" points on various lines using common dressmaker pins. The level of generalization desired was indicated by the number of pins provided. As mentioned previously critical points are those where information regarding the nature of the line is concentrated. They are difficult to identify or rank by mathematical means; however, there seems to be close agreement between both cartographers and non-cartographers in choosing such points.

White carried out research similar to Marino to obtain a data base of critical points obtained from various lines chosen by human beings and performed rigorous statistical tests on four line generalization algorithms. The comparisons made use of 1) area offset: the total space enclosed between the original base line and the generalized lines and 2) common vs uncommon points: the number of points held in common between the base line and the generalized line. The second method was further analyzed by comparing the points picked by the algorithms with those judged "most significant" by the human compilers.

Recently McMaster (1983) has investigated analytical techniques to quantitatively analyze line generalization algorithms. From various sources he collected 30 possible mathematical measures of line generalization quality. These included measures of linear attributes which are applied to single lines and compared between lines, and measures of linear displacement which are applied to two lines. Measures of linear attributes include line length data, coordinate data, angularity data and curvilinearity data. Linear displacement measures include vector difference data, polygon difference data, and perimeter areal polygon data.

These 30 measures were analyzed using correlation coefficients, principal components analysis, and cartographic judgment to identify 6 which were largely statistically independent and which amongst themselves contained all the information obtainable by using any of the 30 measures. The six included:

- Ratio of the change in the number of coordinates - This measure provides a useful standardization in making comparisons across lines.

- Ratio change in the standard deviation of the number of coordinates per inch - This measure indicates whether a generalized line has a uniform coordinate density in relation to the original line.

- Ratio of the change in angularity - This measure evaluates the sum of the angular changes between a line and its generalization.

- Total vector displacement per inch.

- Total areal displacement per inch - Both of the above measures evaluate the displacement between the original line and its generalization.

- Ratio change in the number of curvilinear segments - This measure was retained largely for cartographic reasons. It was hypothesized that the change in number of curvilinear segments is important in evaluating algorithms.

The most complex algorithm tested by White or McMaster was the tolerancing algorithm developed by Douglas and Poiker (1973). Other methods tested included simple n-th point selection, selection by a local angle measurement and selection by a local perpendicular distance measurement.

The Douglas-Poiker algorithm proved superior in both approximating the lines obtained by manual generalization and in approximating the original curves. While this was a useful experiment it does not provide final answers. First, only a small number of algorithms were tested. Many good algorithms were not examined. Second, only isolated lines were studied. A more complex test would need to be based on perception of a whole map. Finally, no provisions were made for taking into account different line weights which might be used to represent the generalized lines at a particular target scale.

More research of this form can be expected in the near future. Several efforts are being carried out in similar areas at a number of universities.

While cartographic testing of line generalization algorithms is limited, testing of displacement algorithms is nonexistent. This is reflected in the much smaller amount of cartographic literature which is available on the displacement problem. It is also shown by the fact that cartographers who were performing similar research a few years ago have apparently moved to other areas. Appendix H discusses contacts with university and research cartographers made to discover the state of their research in the map generalization and displacement field.

7.0 CONCLUSIONS

This chapter presents conclusions on the utilization of the line generalization and feature displacement algorithms described in chapters 4 and 5.

7.1 LINE GENERALIZATION ALGORITHMS

Line generalization algorithms may be judged individually and by categories. The strengths and weaknesss of the nine major categories of line generalization algorithms are discussed below. The categories are also ranked on subjective criteria related to their potential for satisfying DMA automation requirements.

7.1.1 Algorithm Strengths and Weaknesses

Each of the algorithm categories described in chapter 4 has individual strengths and weaknesses. The most important are presented in Table 7.1. When using this table it should be remembered that algorithm performance is a function both of the goals of the compiler and of the input data. Thus an algorithm characteristic which is in general judged to be a strength or weakness may in fact be the opposite in certain unusual conditions.

7.1.2 Line Generalization Ranking

The nine categories of line generalization algorithms can be separated into groups according to their potential use in an automated cartographic environment. Of course any grouping must reflect the criteria used in making the evaluations. The results of ZYCOR's ranking are shown in Table 7.2 Further details of the ranking are given in the following sections.

Table 7.1
Algorithm Strenths and Weaknesses by Algorithm Category

Algorithm Category	Strength	Weakness
Selection	1. Good for thinning oversampled data. 2. Easy to understand	1. Difficult to correlate with cartographic goals 2. May miss significant points 3. Difficult to choose correct thinning rate
Low Pass Filtering	1. Good at removing noise and fine detail. 2. Extensive theory from signal processing to use	1. Cartographically suspect. 2. Smoothes significant points 3. Uniform shrinking of all convex features. 4. Difficult to determine appropriate filter width, overlap, and weighting function. 5. Produces angular results
Angle Detection	1. Finds significant points 2. Many variations may be used to fine tune it	1. Possibly difficult to control. 2. Can find non-significant points 3. Difficult to control accuracy in long regions of low curvature. 4. Produces angular results
DEM Smoothing	1. Useful only for contours 2. Many parameters which may be used for control 3. Requires storage of only one data format for all terrain 4. Allows bathymetric contours to move only seaward. 5. Ties to drains, ridges, etc. 6. Contour output may be at any spacing desired	1. Demands existence of DEM and contouring software 2. Produces completely new contours 3. May move contours substantially in regions of flat terrain 4. Useful only for contour data 5. Non-intuitive control. 6. Correlation between DEM parameters and contour smoothness is unclear.
Tolerance Bands	1. Finds points close to significant points 2. Intuitive control 3. Local versions may be able to detect and remove spikes	1. May find non-significant points 2. Produces angular results 3. May produce spikes
Point Relaxation	1. Intuitive 2. Emulates cartographer creating pull up in some ways 3. Finds points close to significant points	1. Biased toward concave sides of curves. 2. May produce angular results 3. May produce spikes 4. May retain non-significant points
Domain Transformation	1. Extensive research in this area from signal processing	1. Non-intuitive control. 2. No good at handling fine detail 3. No good way to handle open curves 4. No obvious correlation between transform coefficients and line generalization rules
Mathematical Fitting	1. Intuitive 2. Complex algorithms find significant points 3. Conic arc fitting may provide significant smoothing 4. May provide basis for feature detection	1. Angular results 2. There are more direct ways of finding significant points 3. Least squares techniques may miss isolated points which are significant 4. All new data points 5. Difficult to guarantee visually satisfactory results
Epsilon Filtering	1. Some intuitive meaning	1. Unclear how to reconcile areas created in basic algorithm 2. Difficult to program basic algorithm 3. Clustering approach non-intuitive and designed for polygonal area smoothing

Table 7.2

Low Potential Algorithms

- Selection
- Domain Transformation

Potential Algorithms

- Angle Detection
- Mathematical Fitting
- Epsilon Filtering
- Low Pass Filtering

High Potential Algorithms

- Tolerance Bands
- Point Relaxation
- DEM Smoothing

7.1.2.1 Methods of Ranking

For this report two measures have been selected as particularily important to DMA automation requirements. The first is potential for cartographic usefulness. The second is ease of control.

By cartographic usefulness, ZYCOR means the potential of matching algorithm performance with specific goals of a compiler performing the generalization operation. If line generalization is to be successfully automated in an integrated system then this matching must be easily made. Algorithms for which such a correlation is difficult or impossible to identify are poor choices for use in line generalization.

Ease of control is related to cartographic usefulness but is also directed at a requirement that these algorithms be usable for given applications by compilers who are not sophisticated mathematicians. If an algorithm is difficult to control even when a compiler understands its general characteristics, its usefulness will be restricted. A simple, possibly simplistic, measure of ease of use is the number and interpretation of parameters which need to be specified.

7.1.2.2 Low Potential Algorithms

Using these evaulation standards the 9 categories are assigned to 3 groups. The first group is composed of algorithms which are poor candidates for use in automated generalization. This includes the selection algorithms and the domain transformation algorithms. Selection is in this category primarily because correlating thinning rates for one of these algorithms with particular cartographic goals is a hit or miss affair.

Domain transformation methods are also poor candidates for line generalization. This is due to their inability to handle local detail and to difficulties associated with the handling of non-closed curves. Also a high degree of mathematical sophistication is usually required to use these techniques correctly.

7.1.2.3 Potential Algorithms

The second group of four algorithm categories in Table 7.2 is composed of techniques which have potential for use in line generalization but also have fundamental problems in their current state of development. Improvements in algorithm control, identification of optimal implementations, and identification of special cases in which they can be expected to perform well would be required before they should be included in a production environment.

Angle detection algorithms are in this group because they are difficult to control and because there are numerous special cases which have been noted in the literature for which their performance degenerates. Although they directly attack the problem of finding critical points their current implementation is not satisfactory.

Mathematical fitting algorithms also have numerous associated difficulties. A major problem is that they seem to attack the segmentation problem backward, arriving at break points from the fit rather than from more direct approaches. It is uncertain that a satisfactory method of control can be developed. Arc fitting methods which respect break points found using other approaches and which satisfy simple maximum tolerances are more promising than least squares techniques using polynomial functions.

Epsilon filtering is in this category because no one has yet suggested a way to resolve the areas created by rolling a ball on opposite sides of a curve. There conceivably may be a place for this algorithm in the generalization of boundary data as is described in Chrisman's 1983 implementation. However the complexity involved with using clustering algorithms makes it difficult to see how this algorithm could be easily controlled.

Low pass filtering is included since, although in many cases it would seem to satisfy requirements for the reduction of fine detail while maintaining major detail, control may be difficult to achieve. Certainly any useful implementation must restrict the number of possible control parameters significantly.

7.1.2.4 High Potential Algorithms

The third group is made up of algorithms which are the most promising for use in line generalization. DEM filtering for contour generalization is included primarily because it provides the most promise of handling features rather than just lines when generalizing contours. DEMs may have enough well defined information content to provide quantifiable guidance to compilers.

Point relaxation and tolerance band algorithms are included because they address an easily understood goal, the finding of critical points, and are easy to control. For these algorithms it is easy to correlate changes in the scale of the single parameter with changes in line shape.

7.1.3 Line Generalization Systems

No existing algorithm will solve all line generalization problems. Nor is such a single algorithm likely to be developed. The multitude of algorithms published for this purpose is itself an indication of the unsettled nature of research in this field. Cartographers and computer scientists can easily develop modifications to old techniques which contain enough innovative ideas to warrant publication. There is little evidence that they are moving closer to identifying approaches which are obviously and generally superior.

A major reason for this failure is that algorithm evaluation is largely a subjective task. If there are clearly superior approaches to this problem they will not be identified through the addition of any number of figures at the end of an article in a journal. They may be identified through an expensive process of experimentation which tests the responses of potential users under controlled conditions.

Whether or not an experimental process is completed, ZYCOR expects that the generalization of linear features will need to be an interactive process for a considerable period. Even when recommended algorithms for particular tasks are used, cartographers will wish to continually review performance on particular features or classes of features with the expectation of changing algorithms or parameters or of subdividing the problem so that different algorithms or parameters can be used over separate areas.

We also must remember that line generalization will be only a part of complete map compilation system which involves complex topological, data base, and AI aspects. These general problems, which include maintaining data consistency during generalization, are not addressed in this report and deserve much study.

It should be noted that the majority of the algorithms discussed in this report are relatively easy to program and that many are easy to use. As long as sufficient guidelines are provided to the compilers there is no reason not to include many algorithms within the capabilities of an automated cartographic system.

7.2 FEATURE DISPLACEMENT

The literature on feature displacement is significantly less than that available on line generalization. Analysis is more a matter of noting gaps in algorithms rather than measuring relative quality. This weakness is caused by the need to aggregate information, something that humans do well and computers do poorly. There would seem to be two fundamental ways of aggregating information positional cartographic data for processing by computer, word map algorithms and AI approaches.

7.2.1 Word Map Algorithms

To perform automated displacement all features in some local area must be identified, their relative importance noted, their individual and collective information content quantified and the conflicts between their position measured. One method of detecting conflicts is to divide the map area into a "word map" in which very small areas or pixels are assigned one or more computer words to record the features which will print over the region covered by the pixel in some placement scheme. Various bits in the word may be used to record feature IDs, sizes and other relative information. If necessary other bits may serve as pointers to arrays which record chains of symbol conflicts, past movements, or conflict histories.

This approach to the displacement problem through the detection of minutely quantized conflicts leads naturally to solutions involving summations of displacement vectors, localized optimization techniques, and small iterative changes. Christ's algorithm is based on such an approach and is the most complete example available in the literature. It leaves major gaps by failing to handle area features and making no provisions for algorithm iteration among other problems.

A major advantage of the word map approach is that with small enough feature movements during a single iteration the problem of topological validity checking may be easily performed With each feature initially placed at its correct geographical position, small movements in a direction which would involve, for example, a building eventually crossing a road may be detected and compensated for in subsequent iterations.

There are also major difficulties with this approach. A primary one is demands on computer resources. A detailed word map with sufficient resolution to support creation of DMA products may easily involve millions of words of computer storage. Localized processing and multilevel resolution in the word map may reduce this problem.

The word map approach does not present obvious means of handling naturally linear or area features. Furthermore it is easy to put together examples which any simple iterative movement scheme will not be able to resolve.

7.2.2 AI Approaches

Christ's algorithm provides the only example of an attempt to provide a general problem solution. All other algorithms described previously are examples of ways to resolve specific problems. To some extent they may emulate the actions of a cartographer dealing with similar problems. As such they are possible tools to be used in complete formulations of the problem from a hueristical or AI approach. None of these techniques can be used in isolation. None of them make any provisions for maintaining and checking topological validity.

If such a fragmented approach to the problem is going to work then data base formulations will play a vital role by providing means to segment the problem, by providing links to adjacent features, and by providing checks on validity of solutions. Putting such a system together from its parts will not be easy.

Existing cartographic data bases consist of collections of point locations; the meaning of these locations is not part of the data base. The user may know that a given file of (X,Y,Z) data represents a gridded map, or that in another file of (X,Y) pairs represents the corners of buildings, but such information tends to be stored not in the data base but externally to the data base - it is implicit in the programs which access the data base, and is explained only in program documentation or in the minds of the programmers who created the system.

Existing commercial data base systems do not have any robust mechanisms by which it would be possible to define the required relationships to support displacement algorithms. There are certain research systems, often called semantic data base management systems, designed specifically to allow users to define and manipulate a variety of relationships between data items as part of the data base definitions. Unfortunately, current implementations are research tools. They are slow, inefficient and can not handle large data bases. Production versions cannot be expected in the near future.

7.2.3 A Displacement Problem Formulation

One big problem with map construction by computers is that there is little known from cartography on the economic consequences of the varying degrees of "goodness" of maps (Bie, 1980). Mathematical models of map quality are needed.

Suppose a measure of map quality exists. What might it be a function of? Two items come to mind easily: Accuracy and legibility or readability. As for accuracy we wish to place features at their correct locations as much as possible. For legibility we demand (among other things) a minimum separation between features. Consider the following problem:

$$\text{MIN} \sum_{i=1}^{N} W_i \left((\hat{x}_i - x_i) \quad + (\hat{y}_i - y_i)^2 \right)^{1/2}$$

subject to

$$(\hat{x}_i - \hat{x}_j)^2 + (\hat{y}_i - \hat{y}_j)^2 > \epsilon^2 \quad \forall i,j$$

Where $\hat{x}_i$ and $\hat{y}_i$ represent the final placement of a feature on the map and x_i and y_i represent its accurate cartographic position. The W's represent weights which adjust the relative importance of accurate positioning between features (feature hierarchies). Thus the problem as presented is one of minimizing total weighted displacement of features subject to constraints on minimal accepted separation between features on the map.

This is a standard mathematical programming problem encountered in economics and business. In many situations this problem has a solution. Iterative techniques for non-linear optimization are widely researched.

However, such a formulation ignores a multitude of real world problems. A purely mathematical problem is that any non-linear optimization problem involving more than 10 or 20 variables is likely to be intractable unless there is a particular structure on the constraints which may be used in the creation of an optimization algorithm designed especially for the problem being considered.

Another problem is that the given formulation is appropriate only for point features. Nor does it contain provisions for dealing with relative feature placement for pattern recognition.