Naval Ocean Research and Development Activity
August 1989 Report 232

Simplification Methods for the World Vector Shoreline

Jerry L. Landrum
Mapping, Charting, and Geodesy Division
Ocean Science Directorate

Foreword

The Naval Ocean Research and Development Activity has been heavily involved with the design of the Defense Mapping Agency's (DMA) World Vector Shoreline (WVS) data product. The completion of worldwide WVS coverage was scheduled for June 1989. A Compact Disk - Read Only Memory Prototype is also planned.

Many Navy systems will be using the WVS. The coastline simplification or thinning methods described will enable the efficient use of the WVS product in mapping applications that require lower levels of resolution. The distribution of lower-resolution coastline data sets by DMA could save individual method development for each system, ensure that all systems display the same coastline at the same map scale, and greatly speed the display.

W. B. Moseley
Technical Director

J. B. Tupaz, Captain, USN
Commanding Officer

Executive Summary

The World Vector Shoreline (WVS) product is being produced at a map scale of 1:250,000. Three algorithms for simplifying or thinning the WVS to lower resolution are implemented, tested, compared, and evaluated. The Douglas-Peucker algorithm is recommended for generating lower resolution data sets for distribution by the Defense Mapping Agency. Other methods have significant speed advantages for real-time applications. Guidelines for selecting a simplification algorithm for a particular application are presented. A resolution threshold of 9 seconds is recommended for map scales from 1:1,000,000 to 1:10,000,000. A resolution threshold of 72 seconds is recommended for map scales smaller than 1:10,000,000.

Acknowledgments

This study was requested by Mr. Mel Wagner, Defense Mapping Agency Systems Center, and sponsored by the Oceanographer of the Navy under Program Element 63704N. Mr. Jim Braud of NORDA assisted with the programming. Mr. Robert E. Mullen, Naval Oceanographic Office, assisted with the development of one of the algorithms and reviewed the manuscript.

Contents

Simplification Methods for the World Vector Shoreline

Introduction

The fractal nature of the coastline[1] makes it desirable to sample the World Vector Shoreline (WVS) data base in a manner related to the map scale at which it will be displayed. A survey of Navy users[2] has determined that several resolutions of the WVS are required.

This report shows the relationships among vector data-base resolution, display device resolution, and map scale, then presents three simplification algorithms. These algorithms provide means of reducing the data-base resolution to that which is appropriate for a particular map scale and display device resolution. The discussion is supplemented with a series of maps illustrating the output of the simplification algorithms at several map scales.

Data Base Resolution, Map Scale, and Display Device Resolution

Digital electronic display devices have a finite resolution. It is not possible to display information in any finer detail. A typical display resolution is 100 pixels per inch, which is used throughout this report. Pixel size in geographic seconds is given by

$$\text{Pixel size} = \frac{3{,}600 \text{ seconds/degree}}{R \text{ (pixels/inch)} * S \text{ (inches/degree)}},$$

where R is the device resolution and S is the map scale.

Table 1 lists pixel sizes for various map scales. The display resolution, R, is 100 pixels/inch.

Pixel size does not necessarily mandate a desirable resolution for all mapping applications. It does offer an objective resolution threshold, since any increase in data-base resolution is wasted as the finer resolution data collapse into the pixel resolution upon display.

Table 1. Map scale vs. pixel size.

Map Scale		Pixel size
Inches/degree	Ratio at equator	(seconds)
16.0	1:270,000	2.25
10.0	1:432,000	3.6
8.0	1:540,000	4.5
4.0	1:1,080,000	9
2.0	1:2,160,000	18
1.0	1:4,320,000	36
0.5	1:8,640,000	72
0.25	1:17,280,000	144
0.1	1:43,200,000	576

Simplification Algorithms

Three simplification algorithms were chosen for implementation, testing, and evaluation. One algorithm is not directly resolution-dependent. The other two algorithms are directly resolution-dependent. Each algorithm produces output consisting of a subset of the input points. The second and third algorithms may be considered to be "excerpolation" algorithms as defined by Sharman.[3]

The first algorithm is simple, but useful, and is known as the *N*th point algorithm. The first and last points in each segment are retained. The intermediate points are selected at some sampling interval, N.

The second algorithm was developed by the author and Robert E. Mullen at the Naval Oceanographic Office in 1985. For this algorithm a resolution threshold is specified in the units of the data. The first two points and the last point are retained. The distance from each intermediary point to an imaginary line through the two previously selected points is compared to the resolution threshold. If the distance exceeds the threshold, then the point is retained. If the distance is less than the threshold, then the point is rejected.

The third algorithm was described by Robinson[4] as follows: "The Douglas-Peucker algorithm allows the cartographer to specify a threshold that controls the amount of simplification. For a specified line segment the two end points are connected by a straight line and the perpendicular distances from all the intervening points to that line are calculated. If a perpendicular distance exceeds the specified threshold, the point with the greatest perpendicular distance is used as the new end point for the subdivision of the original line. The perpendicular distances from all the intervening points are calculated and compared to the specified threshold. If at any time none of the perpendiculars exceed the threshold, all of the intervening points are eliminated. The routine continues until all possible points have been eliminated."

The advantages and disadvantages of the algorithms are summarized in Table 2.

The three algorithms appear in FORTRAN 77 coded form in the appendix.

Map Description

From Table 1, resolutions of 9 seconds and 72 seconds were chosen to test and evaluate the simplification algorithms and as likely candidates for useful

Table 2. Algorithm advantages and disadvantages

Algorithm	Advantages	Disadvantages
*N*th point	Uniform sampling rate. Plots look good. Very fast and simple.	No guarantee of spacial resolution. Directionally dependent.
Landrum-Mullen	Guarantees spatial resolution. Moderately fast. Simple.	Always keeps the first two points. Variable point spacing. Directionally dependent.
Douglas-Peucker	A standard published algorithm. Guarantees spatial resolution with minimum number of points.	Very slow. Variable point spacing. Not simple to implement.

Table 3. Algorithm performance statistics.

Algorithm	Resolution	Vertices % of orig.	File size % of orig.	Run time minutes
*N*th point	*N* = 5	25	48	42
*N*th point	*N* = 20	11	42	42
Landrum-Mullen	9 seconds	29	55	50
Landrum-Mullen	72 seconds	13	41	46
Douglas-Peucker	9 seconds	20	47	76
Douglas-Peucker	72 seconds	10	40	70

lower-resolution data sets. The 9-second resolution would be appropriate for high-quality maps at scales from 1:1,000,000 to 1:10,000,000. The 72-second resolution would be appropriate for high-quality maps at scales less than 1:10,000,000. For the *N*th point algorithm, *N* was chosen at 5 and 20 to produce approximately equivalent resolutions. A sample WVS data set was processed using the three simplification algorithms and the two chosen resolutions to produce six simplified WVS files.

Table 3 presents the results of the six simplifications performed. For each simplification and resolution, the percentage of the original vertices remaining after simplification, the percentage of the original file size, and the run time in minutes are listed. An estimated 40 minutes of the run time was spent on input and output rather than simplification processing.

The reduction in file size is non-linear because all of the attribute information associated with the coastline segments is retained and no segments are completely eliminated.

The full-resolution data file and the six simplified files are plotted as follows:

Figures 1-7	0.1 inch/degree	(approx. 1:43,200,000)
Figures 8-14	1.0 inch/degree	(approx. 1:4,320,000)
Figures 15-21	10.0 inch/degree	(approx 1:432,000)

Each of these figure groups shows an order of magnitude zoom on the upper-left corner of the previous figure group.

Map scale = 0.1 inch/degree (approx. 1:43,200,000)

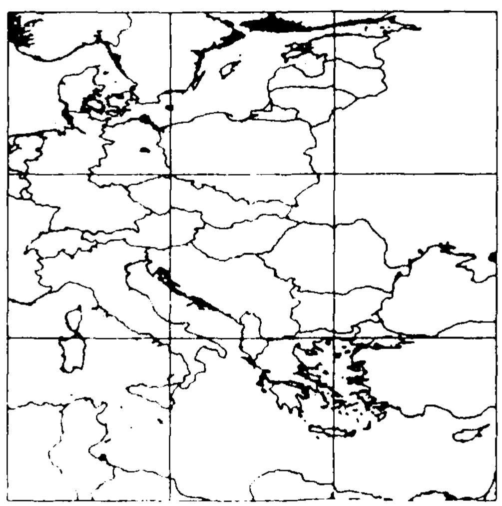

Figure 1. WVS at original resolution.

Observations

1. At this small map scale all of the simplified coastlines appear as good or better than the original, even though the number of vertices is reduced to as little as 10 percent of the original.

Map scale = 0.1 inch/degree (approx. 1:43,200,000)

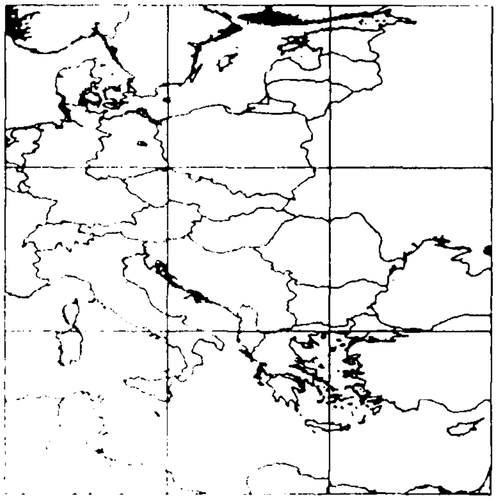

Figure 2. Nth point, N = 5.

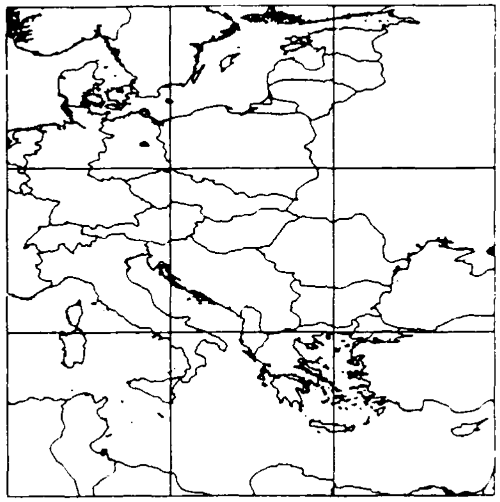

Figure 3. Nth point, N = 20.

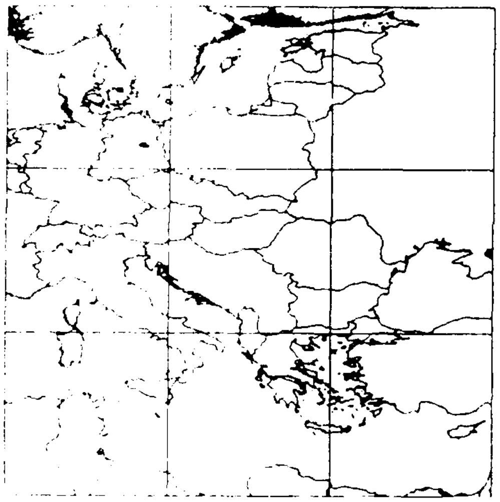

Figure 4. Landrum-Mullen, 9 sec.

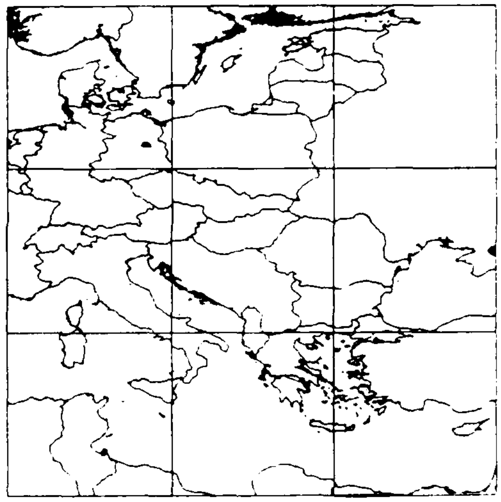

Figure 5. Landrum-Mullen, 72 sec.

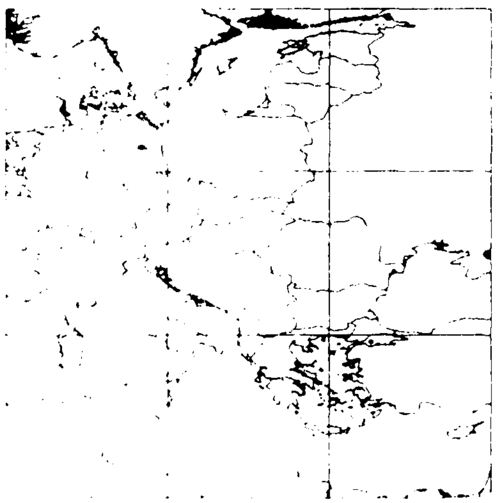

Figure 6. Douglas-Peucker, 9 sec.

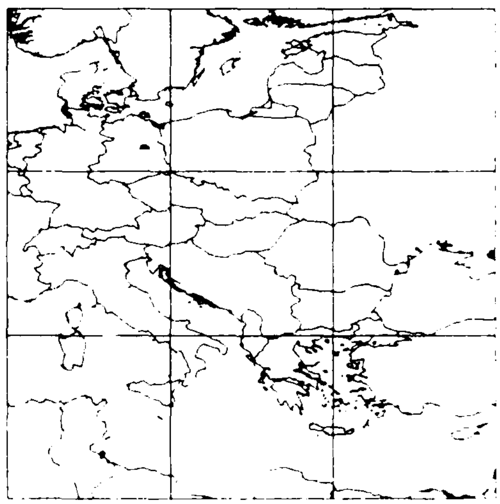

Figure 7. Douglas-Peucker, 72 sec.

Map scale = 0.1 inch/degree (approx. 1:4,320,000)

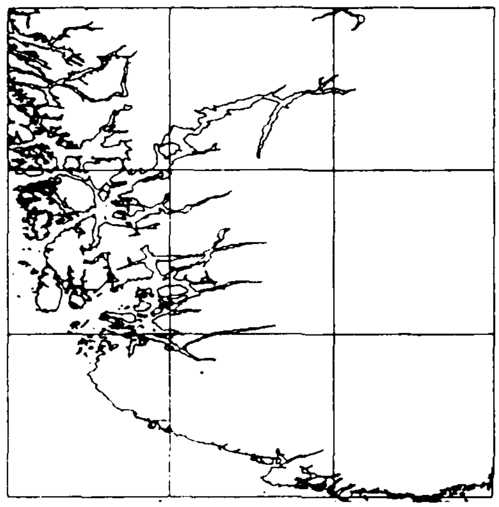

Figure 8. WVS at original resolution.

Observations

1. At this map scale the 9-second, or $N = 5$, resolution is appropriate for high-quality maps.

2. The 72-second, or $N = 20$, resolution may be adequate for some applications.

Map scale = 1.0 inch/degree (approx. 1:4,320,000)

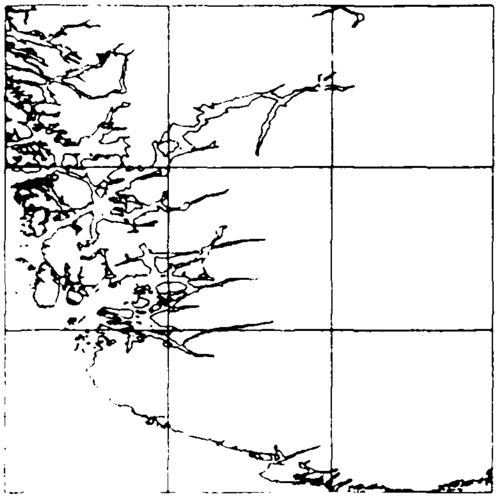

Figure 9. Nth point, N = 5.

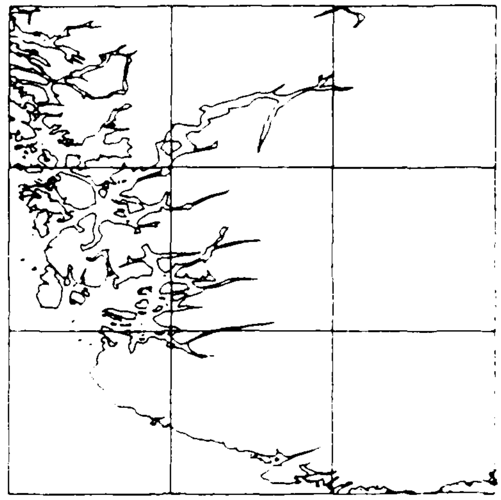

Figure 10. Nth point, N = 20.

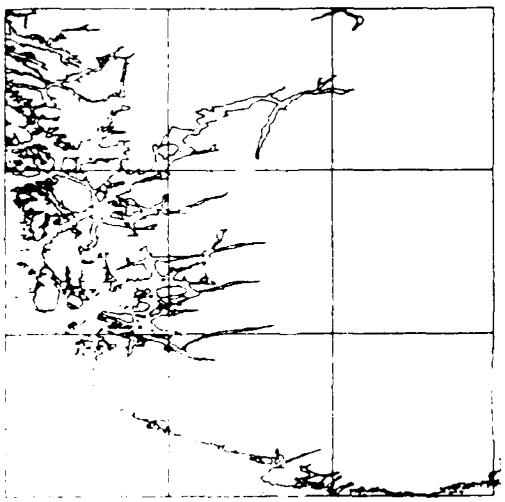

Figure 11. Landrum-Mullen, 9 sec.

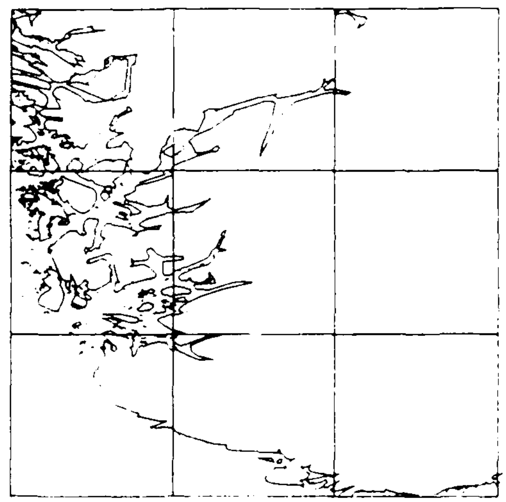

Figure 12. Landrum-Mullen, 72 sec.

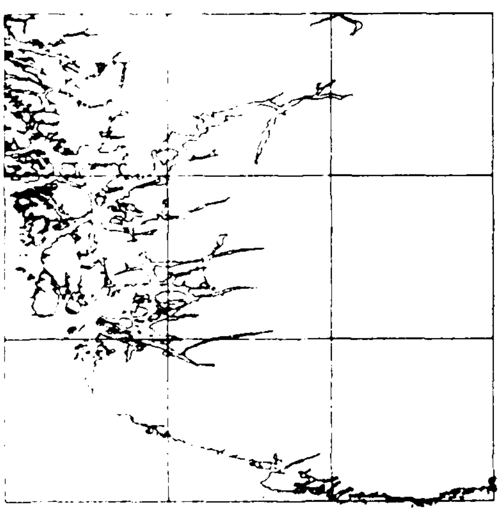

Figure 13. Douglas-Peucker, 9 sec.

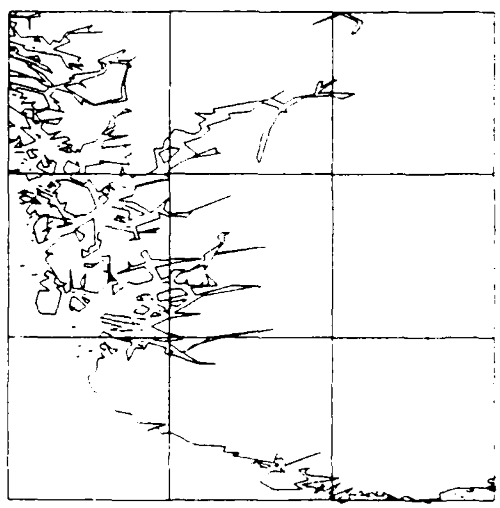

Figure 14. Douglas-Peucker, 72 sec.

Map scale = 10.0 inch/degree (approx. 1:432,000)

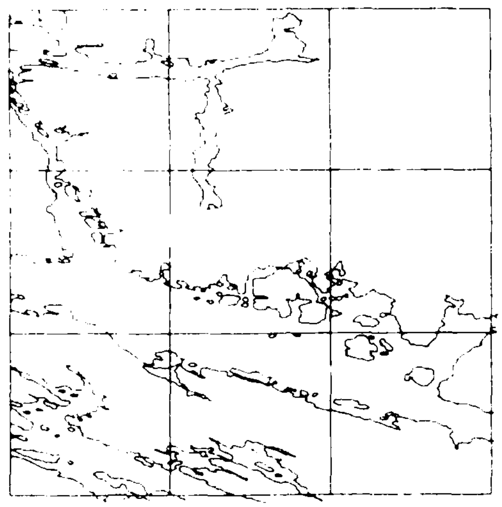

Figure 15. WVS at original resolution.

Observations

1. At this large map scale the original resolution is appropriate for high-quality maps.
2. The 9 second, or N [illegible] 5, resolution may be appropriate for some applications.
3. This map scale emphasizes the differences between the simplification algorithms.

Map scale = 10 inch/degree (approx. 1:432,000)

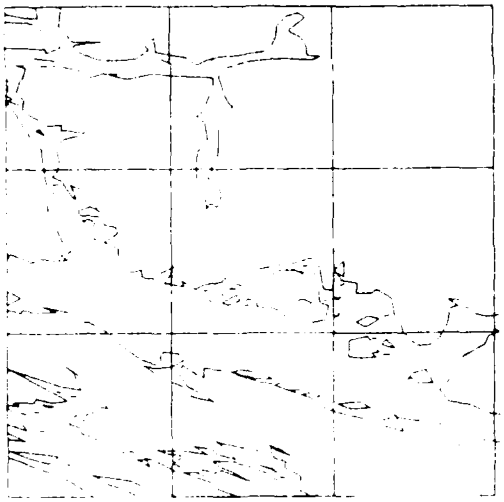

Figure 16. Nth point, N = 5

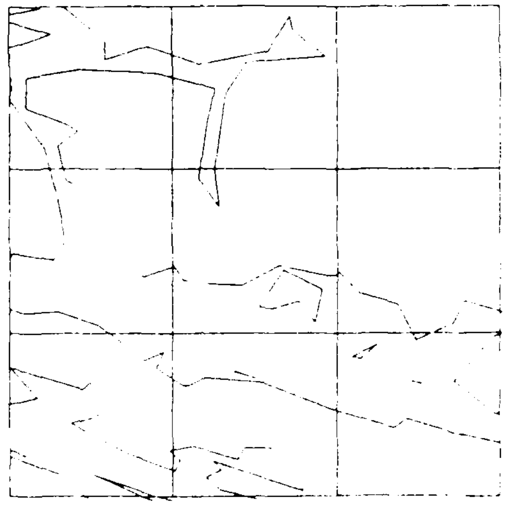

Figure 17. Nth point, N = 20

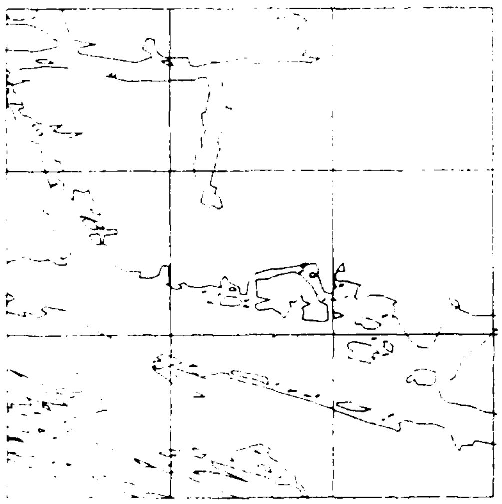

Figure 18. Landrum-Mullen, 9 sec.

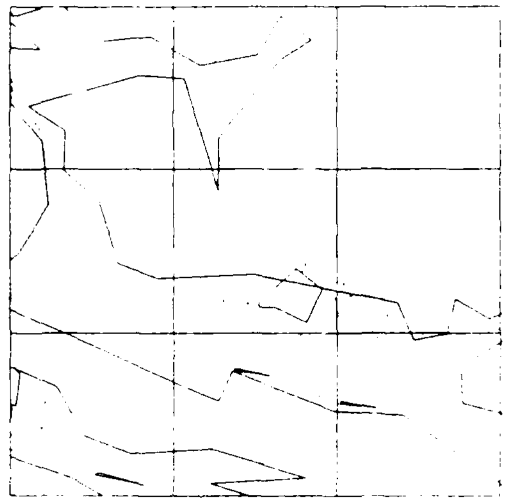

Figure 19. Landrum-Mullen, 72 sec.

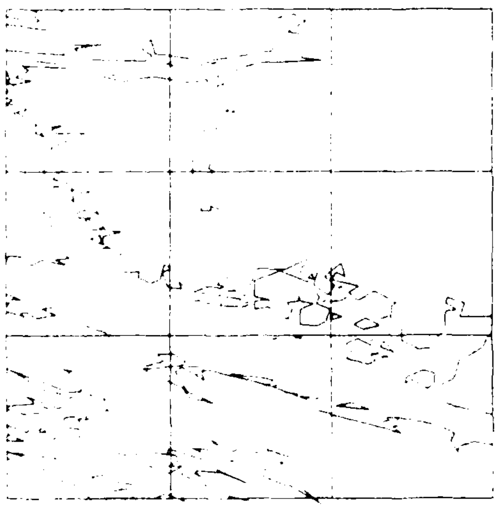

Figure 20. Douglas-Peucker, 9 sec.

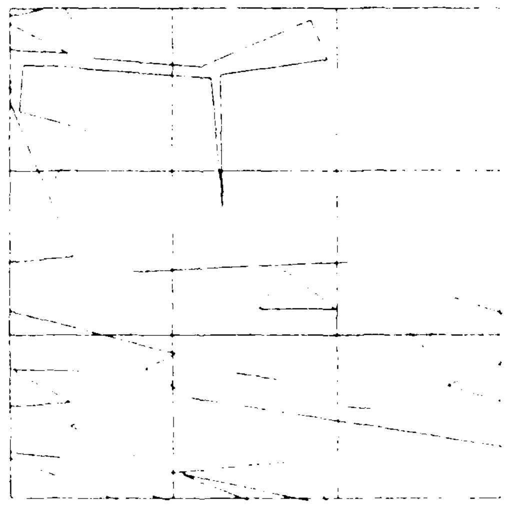

Figure 21. Douglas-Peucker, 72 sec.

Conclusions

• A number of good algorithms are available for simplifying high-resolution vector data bases as a preprocessing step to their display on digital systems at lower resolution.

• When data bases are simplified to the appropriate resolution, the number of vertices is reduced; reduced storage requirements and faster display are the results.

• Generally, the better the simplification algorithm, the more complex and time consuming it will be.

Recommendations

• If processing speed is most important, then the Nth point and Landrum-Mullen algorithms are recommended.

• If maintaining a specified positional accuracy is most important, then the Landrum-Mullen and Douglas-Peucker algorithms are recommended.

• If minimizing the number of vertices while maintaining a specified positional accuracy is most important, then the Douglas-Peucker algorithm is recommended.

• For the one-time production of simplified World Vector Shoreline data bases for distribution by DMA, the Douglas-Peucker algorithm would probably be most desirable.

• All of the simplification algorithms should be made available to the WVS users so that choices can be made for system optimization and error budgeting.

• The issue of feature elimination (the total elimination of features that simplify down to a threshold size) should be resolved on a case-by-case basis depending on the application.

References

1. Mandelbrot, Benoit B. (1977). *The Fractal Geometry of Nature. W. H. Freeman and Company*, New York, NY.

2. Lohrenz, Maura C. (1988). *Design and Implementation of the Digital World Vector Shoreline Data Format*, Naval Ocean Research and Development Activity, Stennis Space Center, MS, NORDA Report 194.

3. Sharman, George F. (1989). Resolution Dependent Data Compression: "Excerpolation", *Proceedings Third Naval Digital MC&G Interest Group Meeting,* Naval Ocean Research and Development Activity, Stennis Space Center, MS. (in prep).

4. Robinson, Arthur H. (1978). *Elements of Cartography*, John Wiley & Sons, New York, NY.

Appendix

Fortran Simplification Programs

APPENDIX A. FORTRAN SIMPLIFICATION PROGRAMS

_A35$DJC1:[DATABASE.WVS]NTHPT.FOR;9 4-APR-198

```
        SUBROUTINE NTHPT(xin,yin,numin,nth,xout,yout,numout)
c  simplifies using Nth point algorithm
c  INPUTS:
c       XIN(numin)- real- array of input x coordinates
c       YIN(numin)- real- array of input y coordinates
c       NUMIN- integer- number of input points
c       NTH- integer- the point interval to be extracted
c  OUTPUTS:
c       XOUT(numin)- real- array of output x coordinates
c       YOUT(numin)- real- array of output y coordinates
c       NUMOUT- integer- number of output coordinates
c  Jerry Landrum, NORDA/DMAP (601) 688-4613, AUTOVON 485-4613

        real xin(numin),yin(numin),xout(numin),yout(numin)
        iin=1
        numout=0
        do 100 i=1,numin,nth
                iin=iin+nth
                numout=numout+1
c       write(*,*) ' i,numout',i,numout
                xout(numout)=xin(i)
                yout(numout)=yin(i)
100     continue
        if(iin.ne.numin) then
c  the loop ended short of last point in segment so include it
                numout=numout+1
                xout(numout)=xin(numin)
                yout(numout)=yin(numin)
        end if
        return
        end
```

```
      REAL FUNCTION DIST3(x1,y1,x2,y2,x3,y3)
c  returns the distance from the point x3,y3 to a line through
c    the two points x1,y1 and x2,y2

      dx=(x2-x1)
      dy=(y2-y1)
c  vertical and horizontal slopes are trivial cases
      if(dx.eq.0.)then
        dist3=abs(x3-x1)
      else if(dy.eq.0.)then
        dist3=abs(y3-y1)
      else
c  From analytic geometry
c  slope and intercept(b) of line connecting (x1,y1) and (x2,y2)
        slope1=(y2-y1)/(x2-x1)
        b1=y1-(slope1*x1)
c  by definition of perpendicular lines, slope from point to line is
        slope2=-(1/slope1)
c  the intercept of the perpendicular is
        b2=y3-(slope2*x3)
c  using substitution we obtain the point of intersection of the line an
c  its perpendicular
        x=(b2-b1)/(slope1-slope2)
        y=((b2*slope1)-(b1*slope2))/(slope1-slope2)
c  the distance we seek is the distance from x3,y3 to the point of inte
        dist3=sqrt((x-x3)**2+(y-y3)**2)
      end if
      return
      end
```

```
      SUBROUTINE LANDRUM(xin,yin,numin,res,xout,yout,numout)
c  Landrum-Mullen algorithm for resolution dependent simplification of a
c       string of points.
c  INPUTS:
c       XIN(numin)- real- array of input x coordinates
c       YIN(numin)- real- array of input y coordinates
c       NUMIN- integer- number of input points
c       RES- real- the desired resolution in the units of the input poin
c  OUTPUTS:
c       XOUT(numin)- real- array of output x coordinates
c       YOUT(numin)- real- array of output y coordinates
c       NUMOUT- integer- number of output coordinates
c  Jerry Landrum, NORDA/DMAP (601) 688-4613, AUTOVON 485-4613

c  Examines the input arrays xin and yin, dropping out points within dis
c  res,
      real xin(numin),yin(numin),xout(numin),yout(numin)
      numout=0.
c  keep first two points
      do 10 i=1,min(numin,2)
        xout(i)=xin(i)
        yout(i)=yin(i)
        numout=numout+1
10    continue

      do 20 i=3,numin-1
        d=dist3(xout(numout-1),yout(numout-1),xout(numout),yout(numout),
     &          xin(i),yin(i))
        if(d.gt.res) then
          numout=numout+1
          xout(numout)=xin(i)
          yout(numout)=yin(i)
        end if
20     continue

c  keep last point
      if(numin.gt.2) then
        numout=numout+1
        xout(numout)=xin(numin)
        yout(numout)=yin(numin)
      end if

      return
      end
```

```
            SUBROUTINE DOUGLAS(xin,yin,numin,res,xout,yout,numout)
c     Douglas-Peuker algorithm for resolution dependent simplification of a
c           string of points. Ref Robinson,A.H.,ELEMENTS  OF CARTOGRAPHY,
c           Wiley,1984
c     INPUTS:
c           XIN(numin)- real- array of input x coordinates
c           YIN(numin)- real- array of input y coordinates
c           NUMIN- integer- number of input points
c           RES- real(for compatibility with other simplification routines
c              -- the point interval to be extracted
c     OUTPUTS:
c           XOUT(numin)- real- array of output x coordinates
c           YOUT(numin)- real- array of output y coordinates
c           NUMOUT- integer- number of output coordinates
c     METHOD:
c           A stack of pointers to sections of the string requiring further
c           analysis is maintained using a stack routine. Selected points ar
c           flaged for output in a local character array YESNO.

c     Jerry Landrum, NORDA/DMAP (601) 688-4613, AUTOVON 485-4613

            parameter (MAXLEN=10000)
            real xin(numin),yin(numin),xout(numin),yout(numin)
            character*1 yesno(MAXLEN)
            integer start,end
            logical prin
            prin=.false.
            yesno(1)='Y'
c     Initialize to keep first and last, can all others
            do 10 i=2,numin-1
                    yesno(i)='N'
10          continue
            yesno(numin)='Y'
c     Begin with the entire string of points
            start=1
            end=numin

c     The following loop exits when the pointer stack is empty, it should n
c       reach completion.
            do 100 k=1,numin
                    dmax=0
                    if(prin) write(*,*) ' start,end',start,end
c     Find farthest point from line containing start and end
                    do 20 i=start+1,end-1
c     Compute distance from point to line containing start and end points
                            d=dist3(xin(start),yin(start),xin(end),yin(end),
     &                              xin(i),yin(i))
                            if (d.gt.dmax) then
                                    dmax=d
                                    index=i
                            end if
20                  continue
                    if (dmax.gt.res) then
c     We have a significant point.  Flag the point, handle the segments.
                            yesno(index)='Y'
c     save pointers to the second segment
                            call push(*300,index)
                            call push(*300,end)
```

```
c  and process the first segment
                              end=index
                     else
c  We do not have a significant point, move to next segment

                              call pop(*200,end)
                              call pop(*200,start)
                     end if
100        continue

200        continue
c  The analysis is complete, move the flaged points to output
           numout=0
           do 210 i=1,numin
                     if(yesno(i).eq.'Y') then
                              numout=numout+1
                              xout(numout)=xin(i)
                              yout(numout)=yin(i)
                              if(prin) write(*,*) ' output:',i,xout(numout),y
                     end if
210        continue
           return

300        continue
           print*,'ERROR Stack length exceeded'
           stop
           end

           SUBROUTINE STACK()
c  Stack push and pop.
c  Call Stack() to initialize stack pointer to 0.
c  Call Push(*label,item) to place an integer on the stack
c  Call Pop(*label,item) to pop an integer from the stack
c    RETURN1 is executed upon overflow and underflow.
           save
           parameter(LENSTK=10000)
           integer istack(LENSTK)
           logical prin
           prin=.true.
           iptr=0
           return

           ENTRY PUSH(*,item)
           iptr=iptr+1
           if(iptr.gt.LENSTK) then
                     if(prin) write(*,*) ' STACK OVERFLOW'
                     RETURN1
           else
                     istack(iptr)=item
                     RETURN
           end if

           ENTRY POP(*,item)
           if (iptr.lt.1) then
                     if(prin) write(*,*) ' Stack Underflow'
                     RETURN1
           else
                     item=istack(iptr)
```

```
                   iptr=iptr-1
c                  write(*,*) ' POP',iptr,item
                   RETURN
             end if
             end
```

Distribution List

Asst Secretary of the Navy
(Research, Engineering & Systems)
Navy Department
Washington DC 20350-1000

Chief of Naval Operations
Navy Department
Washington DC 20350-1000
Attn: OP-02
OP-71
OP-0962X, R. Feden
OP-987

Oceanographer of the Navy
Chief of Naval Operations
U.S. Naval Observatory
34th & Mass Ave., NW
Washington DC 20390-1800
Attn: OP-96

Commander
Naval Air Development Center
Warminster PA 18974-5000

Commanding Officer
Naval Coastal Systems Center
Panama City FL 32407-5000

Commander
Space & Naval Warfare Sys Com
Washington DC 20363-5100

Commanding Officer
Naval Environmental Prediction
Research Facility
Monterey CA 93943-5006

Commander
Naval Facilities Eng Command
Naval Facilities Eng Command HQ
200 Stovall St.
Alexandria VA 22332-2300

Commanding Officer
Naval Ocean R&D Activity
Stennis Space Center MS 39529-5004
Attn: Code 100
Code 105
Code 115
Code 117, J. Hammack
Code 125EX
Code 125L (13)
Code 125P (1)
Code 200
Code 300
Code 350, D. Hickman
Code 351, J. Byrnes
Code 351, J. Braud
Code 351, M. Lohrenz

Brooke Farquhar
NORDA Liaison Office
Crystal Plaza #5, Room 802
2211 Jefferson Davis Hwy.
Arlington VA 22202-5000

Commanding Officer
Naval Research Laboratory
Washington DC 20375

Commander
Naval Oceanography Command
Stennis Space Center MS 39529-5000

Commanding Officer
Fleet Numerical Oceanography Center
Monterey CA 93943-5005

Commanding Officer
Naval Oceanographic Office
Stennis Space Center MS 39522-5001
Attn: Code GGAP, B. Mullen
Code A, J. Depner

Commander
Naval Ocean Systems Center
San Diego CA 92152-5000

Commanding Officer
ONR Branch Office
Box 39
FPO New York NY 09510-0700

Commander
David W. Taylor Naval Research Center
Bethesda MD 20084-5000

Commander
Naval Surface Weapons Center
Dahlgren VA 22448-5000

Commanding Officer
Naval Underwater Systems Center
Newport RI 02841-5047

Superintendent
Naval Postgraduate School
Monterey CA 93943

Director of Navy Laboratories
Rm 1062, Crystal Plaza Bldg 5
Department of the Navy
Washington DC 20360

Officer in Charge
New London Laboratory
Naval Underwater Sys Cen Det
New London CT 06320

Director
National Ocean Data Center
WSC1 Room 103
6001 Executive Blvd.
Rockville MD 20852
Attn: G. W. Withee

Director
Woods Hole Oceanographic Inst
P.O. Box 32
Woods Hole MA 02543

University of California
Scripps Institute of Oceanography
P.O. Box 6049
San Diego CA 92106

Officer in Charge
Naval Surface Weapons Center Det
White Oak Laboratory
10901 New Hampshire Ave.
Silver Spring MD 20903-5000
Attn: Library

Commanding Officer
Fleet Anti-Sub Warfare Training Center,
Atlantic
Naval Station
Norfolk VA 23511-6495

Defense Mapping Agency Sys Cen
12100 Sunset Hill Rd. #200
Reston VA 22090-3207
Attn: SGWN
Mel Wagner
Ed Danford

Office of Naval Technology
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 20, Dr. P. Selwyn
Code 228, Dr. M. Briscoe
Code 234, Dr. C. V. Votaw

Office of Naval Research
800 N. Quincy St.
Arlington VA 22217-5000
Attn: Code 10
Code 10D/10P, Dr. E. Silva
Code 12
Code 112, Dr. E. Hartwig

Commander
Naval Sea Systems Command
Naval Sea Systems Command HQ
Washington DC 20362-5101

Commanding Officer
Naval Civil Engineering Laboratory
Port Hueneme CA 93043

Commander
Naval Air Systems Command
Naval Air Systems Command HQ
Washington DC 20361-0001

Pennsylvania State University
Applied Research Laboratory
P.O. Box 30
State College PA 16801

University of Texas at Austin
Applied Research Laboratories
P.O. Box 8029
Austin TX 78713-8029

Johns Hopkins University
Applied Physics Laboratory
Johns Hopkins Rd.
Laurel MD 20707

University of Washington
Applied Physics Laboratory
1013 Northeast 40th St.
Seattle WA 98105

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. Agency Use Only *(Leave blank).*

2. Report Date.
August 1989

3. Report Type and Dates Covered.

4. Title and Subtitle
Simplification Methods for the World Vector Shoreline

5. Funding Numbers.
Program Element No 63704N
Project No. 01987
Task No. 300
Accession No. DN257086

6. Author(s).
Jerry L. Landrum

7. Performing Organization Name(s) and Address(es).
Ocean Science Directorate
Naval Ocean Research and Development Activity
Stennis Space Center, Mississippi 39529-5004

8. Performing Organization Report Number.
NORDA Report 232

9. Sponsoring/Monitoring Agency Name(s) and Address(es).
CNO (OP 096)
U. S. Naval Observatory
34th and Massachusetts Ave
Washington, D.C. 20392-1800

10. Sponsoring/Monitoring Agency Report Number.

11. Supplementary Notes.

12a. Distribution/Availability Statement.
Approved for public release; distribution is unlimited. Naval Ocean Research and Development Activity, Stennis Space Center, Mississippi 39529-5004.

12b. Distribution Code.

13. Abstract *(Maximum 200 words).*

The Naval Ocean Research and Development Activity has been heavily involved with the design of the Defense Mapping Agency's (DMA) World Vector Shoreline (WVS) data product. The completion of worldwide WVS coverage was scheduled for June 1989. A Compact Disk-Read Only Memory Prototype is also planned.

Many Navy systems will be using the WVS. The coastline simplification or thinning methods described will enable the efficient use of the WVS product in mapping applications that require lower levels of resolution. The distribution of lower-resolution coastline data sets by DMA could save individual method development for each system, ensure that all systems display the same coastline at the same map scale, and greatly speed the display.

14. Subject Terms.
mapping, World Vector Shoreline, thinning, map simplification

15. Number of Pages.
19

16. Price Code.

17. Security Classification of Report.	**18. Security Classification of This Page.**	**19. Security Classification of Abstract.**	**20. Limitation of Abstract.**
Unclassified	Unclassified	Unclassified	None

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102