

UNITED STATES DEPARTMENT OF THE INTERIOR
Geological Survey

Conversion of raster image data to vector cartographic data

by

Valerie Paskevich¹

Open-File Report 91-304

This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey Editorial standards. Use of tradenames is for purpose of identification only and does not constitute endorsement by the U.S. Geological Survey.

May 15, 1991

¹Woods Hole, Ma. 02543

CONTENTS

| | |
|--------------------------|----|
| Abstract | 1 |
| Introduction | 1 |
| Preparing the image file | 2 |
| Image Conversion | 4 |
| UNIX processing | 5 |
| Projecting the data | 6 |
| Additional possibilities | 9 |
| Appendix A | 11 |
| Appendix B | 12 |
| Appendix C | 13 |
| References Cited | 15 |

| | |
|---|----|
| Figures | |
| 1. SeaBeam bathymetry as an 8-bit grayscale image. | 3 |
| 2. West Florida Escarpment bathymetry contours. | 7 |
| 3. 8-bit grayscale photo. | 9 |
| 4. Grayscale image converted to 25dn levels. | 10 |
| 5. Smoothed grayscale image converted to 25dn levels. | 10 |

Abstract

This report contains a general description of how to convert a rasterized image to vector file information, such as contour data, for cartographic plotting. The procedure is accomplished by executing computer system specific programs and scripts that are specific to the computer systems installed within this Branch of the United States Geological Survey. The generalized procedures may be a useful example that can be modified by other computer sites. A detailed example, with illustrations, of processing a specific data set is included.

Introduction

The capability to process and create vector data information from rasterized image data has been accomplished utilizing two computer systems available at the Branch of Atlantic Marine Geology. The two computers utilized were: 1- a Digital Equipment Corporation's (DEC) MicroVAX-II with VMS operating system; and 2- a DECstation 3100 with the DEC ULTRIX operating system.

The MicroVAX-II computer system is running the Mini Image Processing System (MIPS) software (Chavez, 1984) developed by the U.S. Geological Survey in Flagstaff, Arizona. The MIPS software has been developed over a number of years as an office research tool for remote sensing applications. Subsequently, during the early 1980's, the MIPS software was expanded to process side-scan sonar data collected by the GLORIA system (Geologic Long-Range Inclined Asdic), a long-range sub-sea acoustic imaging device developed by the Institute of Oceanographic Sciences, Wormley, England. The MIPS software has been used extensively by the Branch to process and digitally mosaic GLORIA data collected in the United State's Exclusive Economic Zone (EEZ) of the Atlantic East Coast, the Gulf of Mexico and Puerto Rico. More recently, the MIPS software has been further expanded to process high-resolution side-scan sonar data and swath bathymetric data and to integrate various other types of marine geologic data.

The second computer system, a DECstation 3100, has the capability to run the U. S. Geological Survey's MAPGEN software (Evenden and Botbol, 1985). The MAPGEN software has been installed on various UNIX based systems. This group of programs is used extensively for transforming digital geographic data to a map. A necessary core to the MAPGEN system is program *proj* (Evenden, 1990). That program is utilized in this procedure to project *x* and *y* coordinates from meters to geographic coordinates.

The purpose of these notes is to provide the user with basic instructions to the programs usage and a practical example to be followed. The example provided is the conversion of a rasterized bathymetric image data file to 100 meter contour line vector files for cartographic plotting. The image file represents a specific geographic area and has been produced at a specific scale and map projection. The example provided takes into account the image parameters. Care should be taken by the user to ensure that the proper cartographic parameters are specified during their processing. To this end, the user should refer to *Map Projections-A Working Manual* (Snyder, 1988) and *An Album of Map Projections* (Snyder and Voxland, 1989) as well as the *proj* manual.

Image processing can require various steps depending up the user's ultimate goal. The user is expected to have some basic knowledge of image processing procedures, the DEC VMS operating system, MIPS, the UNIX operating system and shell programming, and the UNIX editor *vi*.

The processing steps required on each system will be described in detail. Once the user has completed processing and enhancement of their selected image, the processing required on the MicroVAX-II computer system is minimal. The processing required on the UNIX computer system is slightly more complicated. That procedure is summarized

in Appendix C as one UNIX script.

Preparing the image file

The first step in converting the rasterized data to vector information requires the user to create an ASCII data file from the image data. It is assumed the user has some basic knowledge of image processing and has completed enhancement of the image. The image used in this example has been processed and enhanced on the Branch's DEC MicroVAX-II computer system with the MIPS software. Presently the image must be either an 8 or 16-bit image. Processing of 32-bit images is not available at this time.

In preparing the image, there are a few thoughts the user should consider. First, do not scale the image data from one bit type to another. In other words, if the original data was recorded in a 16-bit dynamic range use the 16-bit data as input. The main purpose of these programs are to help you produce contour line information and you will want the contour line information relative to the original data range.

In addition to scaling, the data values must be positive and the user should not invert the image data. To compute proper contour lines the input data must have the larger values (e.g. 2000, 2250) represent the deeper areas while the smaller values (e.g. 250,500) must represent the shallow areas.

Another suggestion is to consider the size of the ASCII output file to be created. The ASCII file can be several times the size of the input file. The larger the image file, the larger the ASCII file created. The user should consider whether the entire file is needed for processing. In many cases, the image file can be scaled down without sacrificing accuracy and detail of the contour data produced.

It may be well to prepare the image by running a small low-pass filter over the image to smooth it. This results in smoother contour lines rather than the jittery lines which may otherwise be produced.

The example data set used is acoustic swath bathymetry collected using the University of Rhode Island's SeaBeam system along the West Florida Escarpment in the Gulf of Mexico. The depth ranges from 1377 meters on the shelf to 3421 meters on the abyssal plain floor along the base of the Escarpment. The SeaBeam data were processed to produce a digital mosaic with 50 meter pixel resolution in an Albers Equal Area projection. The geographic bounds of the mosaic are 25° to 25° 30" north and 84° 45" to 85° west.

The original data had some gaps between adjacent Seabeam survey lines. As part of the image preparation, a small low-pass filter was applied to the image to fill in the localized gaps. Filtering was not done to fill the large gaps because this would have generated artifacts in the image data and subsequently the contour information.

Though the original 16-bit image was used as the example data set, the 16-bit image was converted to 8-bit to allow the inclusion of a gray scale representation of the acoustic swath bathymetry and is shown in figure 1. The 16-bit data were compressed from 1300 to 3425 meters resulting in each `dn_value` of the 8-bit image representing 8.3 meters. The black areas of the image show where no valid data exists.

The bathymetry data, as it is presented in figure 1, shows little detail. Approximately one third of the data ranges between 2100 and 3200 meters. That range of 1100 meters is represented as gray to white. Between the depths of 3200 to 3421 meters, the image contains another third of the data. That depth range of approximately 225 meters is shown entirely in white. Generally an image such as this would produce more visual detail by color coding various `dn` ranges. As an 8-bit gray scale image, considerable detail is lost in the range of 2000 to 3300 meters which is found within the escarpment and canyons. Converting this information to contour line information and plotting the vector contour information shows easily the amount of detail and information that was lost by viewing the data in its gray scale format.



Figure 1. SeaBeam bathymetry as 8-bit grayscale image.

Image Conversion

At the Branch of Atlantic Marine Geology, the majority of image processing and enhancement work is done utilizing the MicroVAX-II computer system with the MIPS software. The image used in this example was processed and prepared on this system. Though cumbersome in regard to data management, it was decided to convert the binary image data to ASCII to eliminate problems which arise from attempting to read binary data on different computer systems. The data records output by program *IMG2CON* contain a maximum of 85 values per record.

To convert the binary image file to its ASCII format, the user must execute the MIPS program *IMG2CON* on the MicroVAX-II computer system. The program is started by entering the following command at the VMS prompt (\$).

```
$ RUN IP:IMG2CON
```

| U. S. G. S. Mini Image Processing System | | | |
|---|---------------------------|--|--------------|
| IMG2CON | Number of Output Files: 1 | Output Data: ASCII | |
| 11-Mar-91 | Number of Input Files: 1 | Input Data: Image: 8,16 | |
| Output File Descriptions | | Input File Descriptions | |
| Disk unit and file name where the results are to be stored. | | Disk unit and file name of the gridded data set to be processed. | |
| Example: ASCII.DAT = DUB0:INPUT.PIC | | | EXIT or HELP |
| FILES>wflbathy.dat=wflbathy.pic | | | |

As shown above, the program will display a typical MIPS program files screen and request the user to input the names of the files to be processed with a FILES> prompt. In this example, the user has specified the input file as *wflbathy.pic* and the output file to be created as *wflbathy.dat*.

As a default function of all MIPS programs, a "print" file is created during execution. Program *IMG2CON* creates a "print" file that contains information the user will need in the UNIX processing. The user should note the number of lines, samples and data range. An example of a typical print file is listed in Appendix A. The user must also note for subsequent use the image pixel resolution, in meters, and the lower left geographic coordinate. That information is not contained in the image file.

Any rasterized or gridded data set may be used in this procedure. If the user does not have access to MIPS or if they utilize a different computer system, he or she may create an ASCII data file for processing by writing a simple conversion program. Even though the data file created by *img2con* stores a maximum of 85 values per record, a data file with one value per record is acceptable for the UNIX processing. The only criterion in creating the ASCII file is that the first value output must be the first data value at the image origin, followed by the values for the first record, and subsequently the remaining image records.

UNIX processing

After the program has completed its processing the user may copy the ASCII data file to the UNIX system for further processing. The UNIX system 'moana' is used in this example. The MicroVAX-II computer system, known as 'mips', has both *DECnet* and *ftp* installed for transferring of data from systems. Since not all UNIX computer systems have *DECnet* available and do have *ftp*, *ftp* was chosen to transfer the data file from system to system. In the following example, user Zaphod Beetlebrox transfers the data file *wflbathy.dat* from mips to moana.

```
$ ftp mips
Connected to whmips.
220 whmips FTP server ready.
Name (mips:zb): zaphodb
Password (mips:zaphodb): xxxxxxx
331 Password required for ZAPHODB.
230 User ZAPHODB logged in.
ftp> get wflbathy.dat
200 PORT command okay.
150 Opening data connection for wflbathy.dat (128.128.19.30,1477)
226 Transfer complete.
local: wflbathy.dat remote: wflbathy.dat
2278080 bytes received in 16 seconds (1.4e+02 Kbytes/s)
```

Once the user has copied the file to the UNIX platform for processing, they may proceed with the creation of the contour line information.

The first program to be executed is program *mips_cont*. This program will read the input file into a memory array and then scan the array to produce a separate file for each contour interval selected. The output file(s) are automatically named as Lnnnnn where nnnnn is the numeric value of the contour interval. In other words, a file named L00250 would contain the information for contour level 250. A typical runline for program *mips_cont* is as follows:

```
mips_cont -i input_file
           -c num_lines,num_samples,pixel_resolution
           -l start_level,contour_interval,end_level
```

The user must specify *-c*, *-l* and *-i* when running program *mips_cont*. The parameters are described as follows:

- i* input file - The name of the input file to be processed.
- c* num_lines - The number of lines/rows contained in the image file.
 num_samples - The number of samples/columns contained in the image file.
- pixel_resolution - The resolution, in meters, that each pixel of the image represents.
- l* start_level - The first contour level to be produced.
 contour_interval - The interval at which the contour are to be produced.
- end_level - The last contour level to be produced.

In addition, the user may specify the option *-q*. The *-q* option tells the program to run in the quiet mode. The pro-

gram default is to run in verbose mode that will output various program parameters to the user's terminal. This information may be of use to the user and may be re-directed to *std_err*. It is suggested that the user use the default option of verbose and check the information provided by the program. An example of the output is contained in Appendix B.

For the example data set, the exact run line is as follows:

```
mips_cont -i wflbathy.dat -c 720,525,50 -l 1500-100-3500
```

When program *mips_cont* begins execution it reads the input file into a memory array. Information from the program is reported to the user's terminal. The initial information tells the user it is allocating memory space for the processing array and the cell size it is processing. This is simply reporting back the user information specified as part of the *-c* specifications. As the data file is read, the line being processed is displayed on the user's screen.

When finished reading the input file, the program displays the message *scanning data* followed by five lines of information. These lines report to the user information which includes the number of grid points and the number of valid points in the data file, the data range and the first and last array locations of valid data. The output is similar to:

```
scanning data
number of grid points: 378000
number of valid points: 229909 ( 60.8%)
Z-range: 1377 - 3421
col range: 2 523
row range: 0 718
```

At this point, the program begins scanning the array to produce the individual contour line files. Program *mips_cont* automatically names the output files being created. The files are named as Lnnnnn where nnnnn is the digital value of the level being processed. For example, the file L01500 contains the information for level 1500. If running in the default verbose mode, program *mips_cont* displays on the user's terminal the name of each file as it is processed. The data file contains the MAPGEN command "#-b" to mark the beginning of individual line segments. The X and Y values contained in the data file are recorded as meters. A small section of file L01500 is shown below.

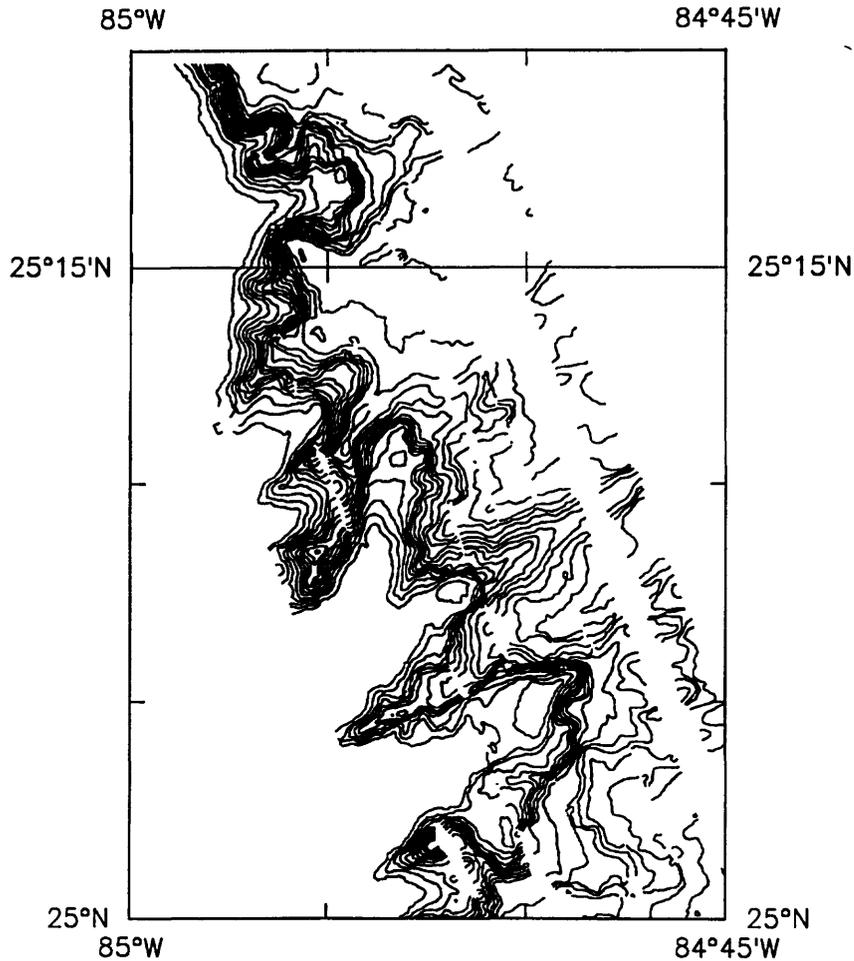
```
moana:233 head -12 L01500
#-b # open
362.8 542
362 541.7
361 541.2
360 541
359.9 541
359 540.4
358.7 540
358 539
357.3 538
357 537.6
356.5 537
```

Projecting the data

The next two steps involve using program *proj* to convert the X/Y meter values to geographic coordinates. It is important that the user have some basic knowledge of the projection requirements before proceeding with these

WEST FLORIDA ESCARPMENT

SeaBeam data converted to contour line data



100 meter contours: 1500 – 3400 meters

Projection: Albers Equal Area

Standard Parallels: 33N & 45N

Spheroid: Clark 1866

Prepared on UNIX MAPGEN system.

Scale 1:250000

5 0 5 KILOMETERS

5 0 5 STATUTE MILES

Figure 2. West Florida Escarpment bathymetry contours.

steps. The first execution of *proj* is to compute the base latitude value. This need only be done once. To compute this value, the user must know the minimum latitude of the area. To compute this value interactively, the user would run *proj* with the following parameters:

```
proj +proj=aea +ellps=clrk66 <<EOF
0 25
EOF
0.00 2577064.35
moana:235
```

After entering the EOF statement, *proj* converts the longitude and latitude coordinates, 0 and 25, to meters. The longitude value may be entered as zero since the purpose is to compute the base_latitude. As shown above, the user should note the returned value for the base_latitude (2577064.35). This value will be needed in the final step of computing the geographic coordinates with *proj*.

The last step of the UNIX processing is to use *proj* again and to project the data files created by *mips_cont*. This step must be executed for every file created by *mips_cont*.

To convert the example data set, the *proj* command line is listed below. This execution of *proj* specifies an inverse projection. *Proj* has many options depending on the projection chosen and will not be described here. The parameters specific to the example data set have been emphasized in bold type.

```
proj +inv +proj=aea +ellps=clrk66 +lon_0=-85 -s -f "%.6f" \
-m 1:50 +y_0=-2577064.35 <L01500 >DL01500
```

The created data file, DL01500, now contains the geographic coordinates to draw the 1500 meter contour interval. A small portion of the final output file is listed below. Note that the geographic coordinates are output with first the latitude coordinate followed by the longitude coordinate.

```
#-b # open
25.247388 -84.821983
25.247252 -84.822375
25.247025 -84.822866
25.246934 -84.823357
25.246934 -84.823406
25.246661 -84.823849
25.246479 -84.823996
25.246023 -84.824340
25.245567 -84.824685
25.245384 -84.824832
25.245111 -84.825078
```

The final data files may now be input to MAPGEN to plot the contour information. The files created for this example data set were plotted and are shown in Figure 2.

Appendix C contains the listing of a UNIX script file to execute the three UNIX procedures. The script file will process each file created by *mips_cont*.

Additional possibilities

The processing described in this report can be applied to any 8 or 16-bit image data. As a final example, figure 3 shows an 8-bit image of a digitized photo. The image was processed to compute vector data at a 25dn interval. The vector data files were then projected and plotted using *mapgen*. Figure 4 shows the results of the image conversion and figure 5 shows the results using the same data file after having applied a small low-pass smoothing filter to the image. The procedure could be applied to side-scan mosaics to create generalized boundaries for image classification, and could also be applied to aerial photos to produce vector information rather than the conventional digitization methods used in the past.



Figure 3. Mickey Mouse.



Figure 4. Mickey converted to 25dn levels.

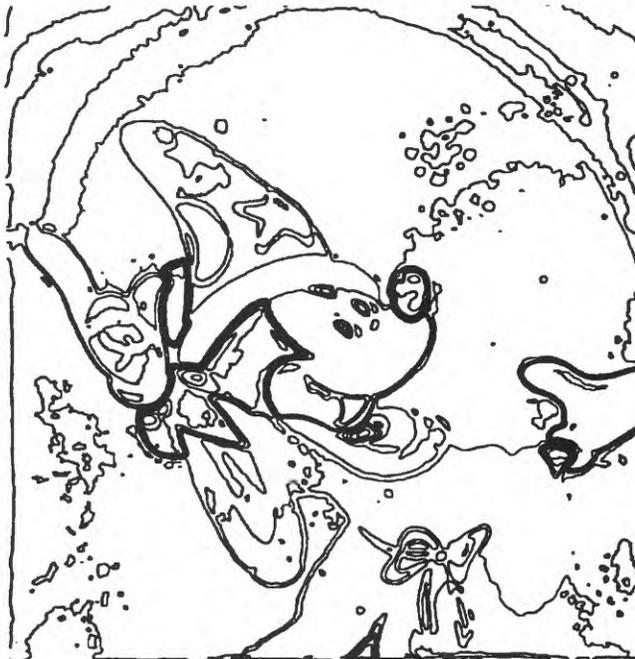


Figure 5. Mickey with smoothing filter applied and converted to 25dn levels.

APPENDIX A

The following is an example "print" file created when running *img2con*.

```
***** IMG2CON *****  
  
Program was run on 12-MAR-91 at 12:50:03  
  
IMAGE data to pre-contour format  
  
INPUT FILE: WFLBATHY.PIC  
  
NL =    720    NS =    525    Bittype = 16  
  
OUTPUT FILE: WFLBATHY.DAT  
  
MINIMUM VALUE output =    1377  
MAXIMUM VALUE output =    3421  
  
The number of zero pixels are 143765  
  
*** END ***    2:32(connect)    1:49.83(cpu)    1009(I/O)    4(faults)
```

APPENDIX B

The following is the run-line used to process the example data set and the processing information output from program *mips_cont*. The percent sign (%) is the UNIX system prompt to the user.

```
% mips_cont -i wflbathy.dat -c 720,525,50 -l 1500-100-3500
```

```
vers 1.1 -- 25 March 1991
```

```
allocating memory for 720 lines/rows 525 samples/cols  
cell size: 50 meters
```

```
loading data  
line/row: 719
```

```
scanning data  
number of grid points: 378000  
number of valid points: 229909 ( 60.8%)  
Z-range: 1377 - 3421  
col range: 2 523  
row range: 0 718
```

```
doing level: 1500 to L01500  
doing level: 1600 to L01600  
doing level: 1700 to L01700  
doing level: 1800 to L01800  
doing level: 1900 to L01900  
doing level: 2000 to L02000  
doing level: 2100 to L02100  
doing level: 2200 to L02200  
doing level: 2300 to L02300  
doing level: 2400 to L02400  
doing level: 2500 to L02500  
doing level: 2600 to L02600  
doing level: 2700 to L02700  
doing level: 2800 to L02800  
doing level: 2900 to L02900  
doing level: 3000 to L03000  
doing level: 3100 to L03100  
doing level: 3200 to L03200  
doing level: 3300 to L03300  
doing level: 3400 to L03400
```

APPENDIX C

The following UNIX script will run the three UNIX procedures as one batch file. This script is written to process the ASCII data file through to the computed geographic coordinates of each contour file. The user must edit the file prior to execution to replace certain values. The values which need to be replaced include pixel resolution, image size (number of lines and samples) and selected contour range and interval and are indicated in bold typeface.

Remember, this script is for an Albers Equal Area projection. If a different map projection is to be used, the user must also make the changes required by *proj* for the new projection.

To execute this script, named *aea-proj*, enter the run-line shown below at the UNIX prompt (%). The two parameters passed to the script represent the lower right geographic coordinate (latitude/longitude) of the image area. This script will automatically compute the base latitude value and pass the information to the final execution of *proj*. In addition, as part of the final execution of *proj*, this script will automatically process every file created by *mips_cont*. For this script to process the *mips_cont* files created, the files must begin with 'L' as described earlier. In the final execution of *proj*, each file is processed and output as 'DLnnnnn' where nnnnn is the level contained within the file.

```
% aea-proj 25 -85

:
# conversion of mips image data to contour data
#
# This file must be edited to substitute the input file name,
# mips_cont parameters and pixel resolution required by proj.
#
# EXECUTE AS: aea-proj base_latitude base_longitude
#
# For example, data file bounds are latitude 25 to 26 and
#                               longitude -84 to -85
#
# run script as:  aea-proj 25 -85
#
# *****
#
# mips_cont options:  -i input file
#                   -c nl,ns,pixres      - grid control information
#                   -l start_contour,contour_level,end_contour
#
mips_cont -i wflbathy.dat -c 720,525,50 -l 1500-100-3500
#
# -----
#
# Let proj compute y_0 base_latitude and substitute value during
# projection.
#
proj +proj=aea +ellps=clrk66 <<EOF >>$t
0 $1
EOF
Y=`cut -f2 $t`
```

```
rm -f $$t
#
# -----
#
# now use proj to convert and project the X/Y data to lat/lon
# -m 1:n specifies pixel resolution in meters
#
for x in L*
do
    proj +inv +proj=aea +ellps=clrk66 +lon_0=$2 -s -f "%.6f" \
        -m 1:50 +y_0=-$Y <$x >D$x
    echo "$x projected"
done
```

REFERENCES

Chavez, Pat S., 1984, U. S. Geological Survey Mini Image Processing System (MIPS), Open-File Report 84-880, 12p.

Evenden, Gerald I. and Botbol, Joseph Moses, 1985, User's Manual for MAPGEN(UNIX version): a method of transforming digital cartographic data to a map, Open-File Report 85-706, 58 pages plus appendixes on font codes and map projections.

Evenden, Gerald I., 1990, Cartographic Projection Procedures for the UNIX Environment - A User's Manual, Open-File Report 90-284, 62 p.

Snyder, John P., 1987, Map Projections - A Working Manual, U. S. Geological Survey Professional Paper 1395, 383 p.

Snyder, John P. and Voxland, R.M., 1989, An album of map projections: U. S. Geological Survey Professional Paper 1453, 249 p.