IMAGE PROCESSING - Rees, Chapter 11

Introduction - Image processing is a vast topic that deserves full course by itself. However, since we have only two lectures, we will focus on the tools that are commonly used to convert satellite remote sensing data into useful information. There are many excellent textbooks on the topic; my favorite is *Digital Image Processing* by Gonzalez and Woods [1992]. These notes basically augment Chapter 11 in Rees [2001], pages 270-296. As noted by Rees, image processing of remote sensing data consists of three steps: *preprocessing*, *image enhancement*, and *image classification*. Before discussing image processing one should understand the way data are stored in a computer file.

Image file formats and metadata - There is a wide range of file formats for remote sensing data and everyone has their favorite. In an ideal world, there would be a standard file format that could be ingested by all software programs. However in the real world, the tool largely depends on the application. Moreover the tools and the data formats constantly evolve while the archived data maintain their original format. A scientist using a variety of data types - both old and new - must somehow cope with all of these formats. Most scientists simply want to read the data into their program without a lot of fussing around with external libraries and complicated code. Can we have both simple code and a standard data format that everyone is happy with? Yes, the solution is to separate the raw data from the information that describes the data.

Consider an image that was acquired along a satellite track as shown in Figure 1. In this satellite co-ordinate system the rows of the data file are perpendicular to the satellite track and columns are parallel to the track.

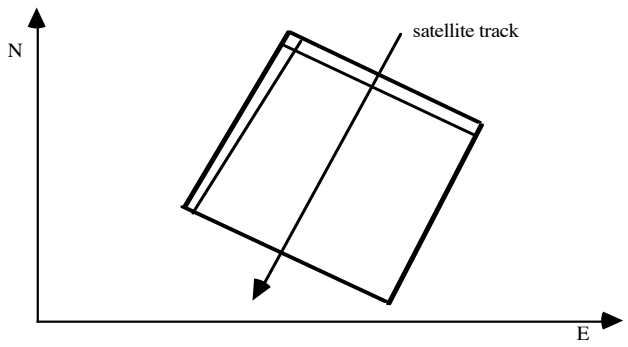


Figure 1. Rows and columns of a raw satellite image before geometric transformation into a longitude, latitude, and height co-ordimate system.

Given precise orbital information along with the orientation of the spacecraft and the characteristics of the sensor, one can develop a formula to map each pixel in the image into latitude, longitude, and elevation. If a precise orbit is not available then ground control points can be used to warp the array from the satellite coordinate system into the ground coordinate system. In any case, after this geo-location step you will still have an array of numbers.

A single-band image is a 2-dimensional array of numbers as shown in Figure 2. The image consisting of M x N pixels, such that the location of each pixel is specified by (i, j). The notation used here (and in most image processing books) is that j represents columns while i represents rows. Note that this is opposite from the notation used in Rees (Figure 11.3).

$$\begin{bmatrix} f(1,1) & f(1,2) & \cdots & f(1,N) \\ f(2,1) & f(2,2) & \cdots & f(2,N) \\ \vdots & \vdots & f(i,j) & \vdots \\ f(M,1) & f(M,2) & & f(M,N) \end{bmatrix}$$

Figure 2. A 2-dimensional matrix of numbers (*pixels*) represents a rectangular satellite image. After geolocation, the first element (1,1) represents the northwest corner of the image. Satellite data are provided in flat binary files that are row-ordered from the top down.

The storage requirement for each pixel will depend on the sensor. For example a single-band image may require just one byte of information (i.e. 8 bits or 0-255) per pixel while a SAR will require two bytes per pixel to represent a complex number. If the instrument has high sensitivity then perhaps the numbers needs to longer (e.g. 2 bytes can represent -32768 to 32767) or maybe floating point numbers are more convenient.

An image acquired by a multi-band sensor produces a 3-dimensional array of numbers. There are three ways of storing the third dimension.

BIP - Band Interleaved by Pixel
BIL - Band Interleaved by Line
BSQ - Band SeQuential

If the pixels of the bands A, B, C and D are denoted a, b, c and d respectively then:

BIP is organized like

```
abcdabcdabcdabcdabcdabcdabcdabcdabcd...   line 1
abcdabcdabcdabcdabcdabcdabcdabcdabcd...   line 2
abcdabcdabcdabcdabcdabcdabcdabcdabcd...   line 3
...
abcdabcdabcdabcdabcdabcdabcdabcdabcd...
abcdabcdabcdabcdabcdabcdabcdabcdabcd...
```

BIL is organized like

```
aaaaaaaaaaaa...   band 1, line 1
bbbbbbbbbbbb...   band 2
cccccccccccc...   band 3
dddddddddddd...   band 4
aaaaaaaaaaaa...   band 1, line 2
...
```

BSQ is organized like and in many cases, `a`, `b`, `c`, and `d` are in separate files.

```
aaaaaaaaaaaa...   line 1, band 1
aaaaaaaaaaaa...   line 2
aaaaaaaaaaaa...   line 3
...
bbbbbbbbbbbb...   line 1, band 2
bbbbbbbbbbbb...   line 2
bbbbbbbbbbbb...   line 3
...
cccccccccccc...   line 1, band 3
cccccccccccc...   line 2
cccccccccccc...   line 3
...
dddddddddddd...   line 1, band 4
dddddddddddd...   line 2
dddddddddddd...   line 3
...
```

As long as each pixel holds the same number of bytes of information and the image array is rectangular then there is a "standard" way of storing the data in two files

header file - The header file is a machine-readable, ASCII file that contains all of the information needed to interpret the raw data file. The computer science people now call this **metadata**. It is hopeless to try to enforce any standard format for the header file, however many systems use a simple **keyword** = **value** structure. *Appendix A* provides 6 example header files that are commonly used today. GeoTIFF is one of the newer formats but, unfortunately, it does not keep the header separate from the data.

data file - The raw data file consists of binary numbers containing the absolute minimum information needed to represent the image. The length of the file in bytes is **bytes_per_pixel** x **number_of_columns** x **number_of_rows**.

Again the header file should contain all the information needed to interpret the data file (*Appendix A*). There are many advantages to this approach.

- To convert from one format to another, only the header file needs to be translated. One could store a variety of header files along with each data file so the data could be used by a variety of software packages.
- One can use the small header files to form a data management system. Indeed, the header files could live on a computer in Dallas while the raw data could sit in an archive in La Jolla and the user could be sitting on a computer in Miami. The user's data browser would point to the header which points to the raw data.
- The lazy scientist can bypass all of this formal stuff. He/she can print out the header, get the basic information, and then write a program that ignores the header and simply reads the raw data. This is what we will do for the Lab next class.

Bit, byte, int, and swap - As discussed above the raw data files contains the minimum number of bytes of data needed to represent the rectangular image. The header file is needed to interpret these raw data. An example of a header file for USGS topography data is:

```
NROWS           6000                - number of rows in file
NCOLS           4800                - number of columns in file
ULXMAP          -99.99583333333334  - longitude of upper-left corner of file
ULYMAP          39.99583333333333   - latitude of upper-left corner of file
XDIM            0.00833333333333    - longitude spacing of pixels
YDIM            0.00833333333333    - latitude spacing of pixels
LAYOUT          BIL                 - band interleaved by line
NBANDS          1                   - number of bands
NBITS           8                   - number of bits per pixel
OFFSET          0.0                 - vertical offset for mapping data number to topography
SCALE           3.92                - scale factor for mapping data number to topography
BANDROWBYTES    4800                - number of bytes per row
```

First consider the parameters related to the horizontal position of each pixel. These are NROWS, NCOLS, ULXMAP, ULYMAP, XDIM, and YDIM. Let *i* be the row number starting from the top and *j* be the column number starting from the left. The following formulas are used to convert the pixel indices (*i, j*) into Earth position (*lat, lon*).

lat = ULYMAP - YDIM * (*i* - 0.5) i = 1, NROWS
lon = ULXMAP + XDIM * (*j* - 0.5) j = 1, NCOLS

The origin of the grid is in the upper left corner, and the grid spacing is 1/120 of a degree. One minute of longitude is 1/60 of a degree so this spacing is 1/2 minute or 30 seconds. This particular mapping from *i, j* to *lon, lat* is called pixel registration and is the most common mapping. The military uses gridline (or node) mapping where the data are defined at points rather that the average over a pixel. If this example had gridline registration instead of pixel registration then there would be the following changes.

NROWS 6001
NCOLS 4801
lat = ULYMAP - YDIM * (*i* - 1) i = 1, NROWS
lon = ULXMAP + XDIM * (*j* - 1) j = 1, NCOLS

Next we consider the mapping of pixel values into real world values. To understand this mapping one really needs to understand the variety of ways that numbers are store on a computer. To begin, consider that transistors in computers have only two states, 1-on or 0-off. This represents one *bit* of information. A *byte* is composed of 8 bits. For example, the byte representation of the number 11 is given in the following table.

128	64	32	16	8	4	2	1
0	0	0	0	1	0	1	1

= 8 + 2 + 1 = 11

In a computer language like C or FORTRAN one would initialize a 1-byte number as:

```
unsigned char   a;
integer*1       a
```

When all the bits are on then the byte represents the number 255. Therefore one byte can be used to represent all of the integers from 0 to 255. Suppose for the moment we have a digital elevation grid of the La Jolla area where the elevations span 0 to 1000 m and we would like to store the

numbers as bytes. We'll call the value of the pixels a *digital number* or DN. The topography can be obtained from the DN using the following formula.

TOPO = SCALE x DN + OFFSET

where

SCALE	1000/255 = 3.92
OFFSET	0.

This works fine, but suppose the initial topography measurements were measured to an accuracy of 0.1 m. Storing the data as one byte retains only a vertical resolution of 3.9 m; this does not capture the full accuracy of the data. In this case a 2-byte representation for the digital number is required. An example of the 2-byte unsigned integer representation of the number 515 is given in the table below.

most significant byte - MSB

32768	16384	8192	4096	2048	1024	512	256
0	0	0	0	0	0	1	0

= 512 + 2 + 1 = 515

least significant byte - LSB

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

In a computer language like C or FORTRAN one would initialize a 2-byte unsigned as:

unsigned short int	a;
unsigned integer*2	a

In this case, one can represent all the integers from 0 to 65535 (i.e., 2^{16}-1). As before, the digital elevations of La Jolla span 0 to 1000 m. Using two bytes per pixel provides a much finer vertical resolution of 1000/65536 =0.0153. In this case, the topography can be obtained from the DN using the following formula.

TOPO = SCALE X DN + OFFSET

where

SCALE	0.0153
OFFSET	0.

The vertical resolution is now more than adequate to capture the full accuracy of the data. We note that it is more common to include a +/- sign in the 2-byte representation of an integer. In this case the number -515 has the bit pattern.

sign	16384	8192	4096	2048	1024	512	256
1	0	0	0	0	0	1	0

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

=-1*(512 + 2 + 1) =- 515

In a computer language like C or FORTRAN one would initialize a 2-byte signed integer as:

```
short int       a;
integer*2       a
```

The header for this 2-byte per pixel representation could be modified (OFFSET) to take full advantage of the range of the 2-byte signed integer.

```
BYTEORDER       M
LAYOUT          BIL
NROWS           6000
NCOLS           4800
NBANDS          1
NBITS           16
BANDROWBYTES    9600
NODATA          -32768
ULXMAP          -99.99583333333334
ULYMAP          39.99583333333333
XDIM            0.00833333333333
YDIM            0.00833333333333
OFFSET          -500.
SCALE           0.0153
```

Suppose we needed even finer vertical resolution, then the data representation could be increased to a signed 4-byte (i.e., 32-bit) integer. The range of these integers is -2,147,483,648 to 2,147,483,647. In a computer language like C or FORTRAN one would initialize a 4-byte signed integer as:

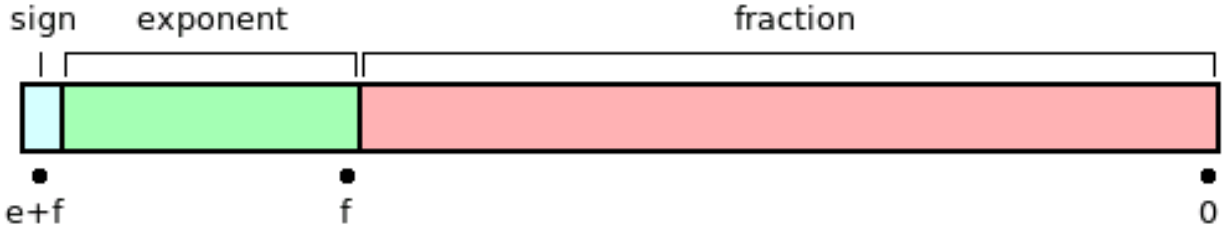
```
long int        a;
integer*4       a
```

In this case we would have a SCALE of 2.328×10^{-7}, probably more than adequate for our La Jolla topography data.

Suppose our data file is small relative to the available memory in our computer and did not want to deal with the SCALE and OFFSET in our computer program. We could more simply store the pixels as 4-byte IEEE floating point values. In a computer language like C or FORTRAN one would initialize a 4-byte floating point number as:

```
float           a;
real*4          a
```

The 32-bit binary representation of a IEEE float is shown in the following diagram.



where the sign takes one bit, the exponent is represented in 8 bits so it can range from 2^{-128} to 2^{127}. In base 10 this corresponds to $10^{-/+38}$. The fraction is represented in 23 bits where 2^{23} = 8388608 which corresponds to about 7 significant digits.

In an ideal world, we could end the discussion here. However, there is one major glitch that will forever plague the world of remote sensing. Two-byte integers are an excellent compromise for data storage - lots of precision but not too large. Unfortunately the engineers (yes blame it on the engineers) came up with **two** ways to store 2-byte integers - big-endian and little-endian. The "right" way to store integers is to order the bytes with the higher byte at a lower memory location as shown in the diagram below for our number -515. In this arrangement, the word can be read from left to right as in a book.

increasing memory address

most significant byte - MSB

sign	16384	8192	4096	2048	1024	512	256
1	0	0	0	0	0	1	0

least significant byte - LSB

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

= -1*(`512 + 2 + 1) = -515

This is called big-endian (or MSB or just M) and it is the way bytes are ordered at Motorola and Sun Microsystems, and Apple Computer and all the "right" places. The little-endian approach has just the opposite byte ordering as

increasing memory address

least significant byte - LSB

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1

most significant byte - MSB

sign	16384	8192	4096	2048	1024	512	256
1	0	0	0	0	0	1	0

= -1*(512 + 2 + 1) = -515

This is the way bytes are arranged by engineers at Intel and AMD and all the backward companies. This is also called LSB or just L. Unfortunately the backward companies are winning and even

Apple has defected to the dark side so almost all computers nowadays have the little-endian representation. Despite the dominance of the little-endian representation, the remote sensing world still uses the big-endian representation. Therefore one needs the ability to *swap* the order of the bytes. At a system level this can be done with the UNIX program dd. For example *dd if=filein of=fileout conv=swab* will swap the 2-byte integers from *filein* to *fileout*. At a program level this can also be done in C. If you do a Google search on endian.h you'll get 87,700 hits! This is a huge problem and the reason why I still use Sun Computers and the power PC Apple CPU's. Big-endian supporters should unite and regain control of this awful situation!!

For the remainder of the Image Processing lecture, I'll follow Chapter 11 in Rees.

Preprocessing (see Rees p. 274-279)

- radiometric correction
- geometric correction

Image enhancement (see Rees p. 279-291)

- contrast modification - linear stretch or histogram equalization
- spatial filtering - smoothing (low-pass), sharpening (high-pass), gradient ((Sobel)

Band transformations (see Rees p. 291-296)

- vegetation index
- principal component transformation

Appendix A - Sample header file formats

1) USGS HDR format used for digital elevation models

Header file “S07W080.hdr” reads as follows:

```
BYTEORDER  M
LAYOUT  BIL
NROWS  1201
NCOLS  1201
NBANDS  1
NBITS  16
BANDROWBYTES  2402
TOTALROWBYTES  2402
BANDGAPBYTES  0
NODATA  -32768
ULXMAP  -80.00000000000000
ULYMAP  -6.0000000000000
XDIM  0.000833333333333
YDIM  0.000833333333333
```

2) USGS NLAPS Header for Landsat Data

```
NDF_REVISION=0.00;
PRODUCT_NUMBER=01196062400240001;
DATA_FILE_INTERLEAVING=BSQ;
TAPE_SPANNING_FLAG=1/1;
START_LINE_NUMBER=1;
START_DATA_FILE=1;
BLOCKING_FACTOR=1;
MAP_PROJECTION_NAME=UTM;
USGS_PROJECTION_NUMBER=1;
USGS_MAP_ZONE=11;
USGS_PROJECTION_PARAMETERS=6378206.400000000400000,6356583.799999999800000,0.0000000000
HORIZONTAL_DATUM=WGS84;
EARTH_ELLIPSOID_SEMI-MAJOR_AXIS=6378137.000;
EARTH_ELLIPSOID_SEMI-MINOR_AXIS=6356752.314;
EARTH_ELLIPSOID_ORIGIN_OFFSET=0.000,0.000,0.000;
EARTH_ELLIPSOID_ROTATION_OFFSET=0.000000,0.000000,0.000000;
PRODUCT_SIZE=FULL_SCENE;
RESAMPLING=CC;
PROCESSING_DATE/TIME=070396/14562000;
PROCESSING_SOFTWARE=NLAPS_2_0_1;
DATA_SET_TYPE=EDC_TM;
PIXEL_FORMAT=BYTE;
PIXEL_ORDER=NOT_INVERTED;
BITS_PER_PIXEL=8;
PIXELS_PER_LINE=6887;
LINES_PER_DATA_FILE=6434;
DATA_ORIENTATION=UPPER_LEFT/RIGHT;
NUMBER_OF_DATA_FILES=7;
LINES_PER_VOLUME=45038;
RECORD_SIZE=6887;
UPPER_LEFT_CORNER=1180447.0227W,0340844.5554N,400455.537,3778643.777;
UPPER_RIGHT_CORNER=1155910.8283W,0335031.4405N,593786.054,3744912.990;
LOWER_RIGHT_CORNER=1162021.0239W,0321254.9927N,562274.270,3564300.847;
LOWER_LEFT_CORNER=1182342.8601W,0323048.8496N,368943.752,3598031.634;
REFERENCE_POINT=SCENE_CENTER;
REFERENCE_POSITION=1171159.6256W,0331101.1264N,481364.903,3671472.312,3444.00,3217.50;
REFERENCE_OFFSET=-0.13,1.26;
ORIENTATION=9.896897;
WRS=040/037.0;
ACQUISITION_DATE/TIME=082790/17422648;
SATELLITE=LANDSAT_5;
SATELLITE_INSTRUMENT=TM;
PIXEL_SPACING=28.5000,28.5000;
PIXEL_SPACING_UNITS=METERS;
PROCESSING_LEVEL=08;
SUN_ELEVATION=53.55;
SUN_AZIMUTH=121.33;
NUMBER_OF_BANDS_IN_VOLUME=7;
BAND1_NAME=TM_BAND_1;
BAND1_WAVELENGTHS=0.45,0.52;
BAND1_RADIOMETRIC_GAINS/BIAS=0.6024314,-1.5200000;
BAND2_NAME=TM_BAND_2;
BAND2_WAVELENGTHS=0.52,0.60;
BAND2_RADIOMETRIC_GAINS/BIAS=1.1750981,-2.8399999;
BAND3_NAME=TM_BAND_3;
BAND3_WAVELENGTHS=0.63,0.69;
BAND3_RADIOMETRIC_GAINS/BIAS=0.8057647,-1.1700000;
BAND4_NAME=TM_BAND_4;
BAND4_WAVELENGTHS=0.76,0.90;
BAND4_RADIOMETRIC_GAINS/BIAS=0.8145490,-1.5100000;
BAND5_NAME=TM_BAND_5;
BAND5_WAVELENGTHS=1.55,1.75;
BAND5_RADIOMETRIC_GAINS/BIAS=0.1080784,-0.3700000;
BAND6_NAME=TM_BAND_6;
BAND6_WAVELENGTHS=10.40,12.50;
BAND6_RADIOMETRIC_GAINS/BIAS=0.0551582,1.2377996;
BAND7_NAME=TM_BAND_7;
BAND7_WAVELENGTHS=2.08,2.35;
BAND7_RADIOMETRIC_GAINS/BIAS=0.0569804,-0.1500000;
END_OF_HDR;
```

3) ERMAPPER Header for Landsat Data

```
DatasetHeader Begin
        Version         = "5.5"
        LastUpdated     = Wed Apr 28 19:22:13 GMT 2004
         SensorName      = "/radar/ermapper/sensortype/Landsat_5.ers"
        SenseDate       = Wed Apr 28 19:22:13 GMT 2004
        DataSetType     = ERStorage
        DataType        = Raster
        ByteOrder       = MSBFirst
        Comments        = "Imported from NLAPS format
NDF_REVISION=0.000000
PRODUCT_NUMBER=01196062400240001
SATELLITE=LANDSAT_5
SATELLITE_INSTRUMENT=TM
ACQUISITION_DATE/TIME=082790/17422648
PROCESSING_DATE/TIME=070396/14562000
PROCESSING_LEVEL=08
PROCESSING_SOFTWARE=NLAPS_2_0_1
RESAMPLING=CC
WRS Path/Row=040/037.0
SUN_AZIMUTH=121.330000
SUN_ELEVATION=53.550000
PRODUCT_SIZE=FULL_SCENE
USGS_PROJECTION_PARAMETERS=6378206.400000000400000,6356583.799999999800000,0.000000000000000,
EARTH_ELLIPSOID_SEMI-MAJOR_AXIS=6378137.000000
EARTH_ELLIPSOID_SEMI-MINOR_AXIS=6356752.314000
X Ellipsoid Offset=0.000000
Y Ellipsoid Offset=0.000000
Z Ellipsoid Offset=0.000000
X ELLIPSOID ROTATION OFFSET=0.000000
Y ELLIPSOID ROTATION OFFSET=0.000000
Z ELLIPSOID ROTATION OFFSET=0.000000
"
        CoordinateSpace Begin
                Datum           = "WGS84"
                Projection      = "NUTM11"
                CoordinateType  = EN
                Rotation        = 9:53:48.8292
        CoordinateSpace End
        RasterInfo Begin
                CellType        = Unsigned8BitInteger
                NullCellValue   = 0
                CellInfo Begin
                        Xdimension      = 28.5
                        Ydimension      = 28.5
                CellInfo End
                NrOfLines       = 6434
                NrOfCellsPerLine        = 6887
                RegistrationCoord Begin
                        Eastings        = 400455.537
                        Northings       = 3778643.777
                RegistrationCoord End
                RegistrationCellX       = -0.5
                RegistrationCellY       = -0.5
                NrOfBands       = 7
                BandId Begin
                        Value           = "0.485000"
                        Width           = 0.07
                        Units           = "um"
                        Comment         = "TM_BAND_1"
                BandId End
                BandId Begin
                        Value           = "0.560000"
                        Width           = 0.08
                        Units           = "um"
                        Comment         = "TM_BAND_2"
                BandId End
                BandId Begin
                        Value           = "0.660000"
                        Width           = 0.06
                        Units           = "um"
```

```
                        Comment         = "TM_BAND_3"
                BandId End
                BandId Begin
                        Value           = "0.830000"
                        Width           = 0.14
                        Units           = "um"
                        Comment         = "TM_BAND_4"
                BandId End
                BandId Begin
                        Value           = "1.650000"
                        Width           = 0.2
                        Units           = "um"
                        Comment         = "TM_BAND_5"
                BandId End
                BandId Begin
                        Value           = "11.450000"
                        Width           = 2.1
                        Units           = "um"
                        Comment         = "TM_BAND_6"
                BandId End
                BandId Begin
                        Value           = "2.215000"
                        Width           = 0.27
                        Units           = "um"
                        Comment         = "TM_BAND_7"
                BandId End
        RasterInfo End
DatasetHeader  End
```

4) Planetary Data Systems (PDS) Header for Magellan Radar Data

```
PDS_VERSION_ID                  = PDS3

/*           FILE FORMAT AND LENGTH */

RECORD_TYPE                      = FIXED_LENGTH
RECORD_BYTES                     =    312
FILE_RECORDS                     =    361
LABEL_RECORDS                    =      9
INTERCHANGE_FORMAT               = BINARY
/*           POINTERS TO START RECORDS OF OBJECTS IN FILE */
^IMAGE                           =     10
/*           IMAGE DESCRIPTION */
DATA_SET_ID                       = "MGN-V-RDRS-5-DIM-MAPMAKER2.0-V1.0"
PRODUCT_ID                       = "MAPMAKER_29S134"
PRODUCER_INSTITUTION_NAME        = "UNITED STATES GEOLOGICAL SURVEY"
PRODUCT_TYPE                     = MDIM
MISSION_NAME                     = "VENUS"
SPACECRAFT_NAME                  = "MAGELLAN"
INSTRUMENT_NAME                  = "RADAR SYSTEM"
TARGET_NAME                      = "VENUS"
FILTER_NAME                      = "N/A"
START_TIME                       = "N/A"
STOP_TIME                        = "N/A"
SPACECRAFT_CLOCK_START_COUNT     = "N/A"
SPACECRAFT_CLOCK_STOP_COUNT      = "N/A"
PRODUCT_CREATION_TIME             = 2004-04-28T12:59:1
/*           DESCRIPTION OF OBJECTS CONTAINED IN FILE */
OBJECT                           = IMAGE
  BANDS                          = 1
  BAND_STORAGE_TYPE               = BAND_SEQUENTIAL
  LINES                          = 352
  LINE_SAMPLES                   = 312
  SAMPLE_TYPE                     = UNSIGNED_INTEGER
  SAMPLE_BITS                    = 8
  SAMPLE_BIT_MASK                 = 2#11111111#
END_OBJECT                       = IMAGE

OBJECT                            = IMAGE_MAP_PROJECTION
  COORDINATE_SYSTEM_TYPE         = "BODY-FIXED ROTATING"
  COORDINATE_SYSTEM_NAME         = "PLANETOGRAPHIC"
  MAP_PROJECTION_TYPE            = "SINUSOIDAL"
  MAP_RESOLUTION                 = 88.000000
  MAP_SCALE                      = 1.200112
  MAXIMUM_LATITUDE               = -27.500000
  MINIMUM_LATITUDE               = -31.500000
  EASTERNMOST_LONGITUDE          = 136.000000
  WESTERNMOST_LONGITUDE          = 132.000000
  LINE_PROJECTION_OFFSET         = 2417.998963
  SAMPLE_PROJECTION_OFFSET       = -158.113840
  A_AXIS_RADIUS                   = 6051.000000
  B_AXIS_RADIUS                   = 6051.000000
  C_AXIS_RADIUS                   = 6051.000000
  FIRST_STANDARD_PARALLEL        = "N/A"
  SECOND_STANDARD_PARALLEL       = "N/A"
  POSITIVE_LONGITUDE_DIRECTION  = EAST
  CENTER_LATITUDE                = "N/A"
  CENTER_LONGITUDE                = 134.000000
  REFERENCE_LATITUDE             = "N/A"
  REFERENCE_LONGITUDE            = "N/A"
  LINE_FIRST_PIXEL               = 1
  SAMPLE_FIRST_PIXEL             = 1
  LINE_LAST_PIXEL                = 352
  SAMPLE_LAST_PIXEL              = 312
  MAP_PROJECTION_ROTATION        = 0.0000000
END_OBJECT                        = IMAGE_MAP_PROJECTION
END
```

5) GIPS Header (author, Peter Ford, MIT) foe global topography of the Earth

```
header = "Image Header"
version = ieee
title = "World Gravity Image"
owner = "David T. Sandwell <sandwell@baltica.ucsd.edu>"
origin = "Geosat and ERS-1 Altimeter Profiles"
date = "Mon Sep 11 07:51:42 PST 1995"
mapping = "name=Mercator type=8 origin=5399.5,3167.5 radius=1718.87 euler=-180,0,0 lat=0,0
"
dtitle = "Gravity Anomaly"
dtype = h
dscale = n
dmax = 2000
dmin = -2000
dfact = 1.0
dzero = .00
xtitle = Longitude
xfont = screen.b.14
xscale = w
xnum = 10800
xorg = 0
xmax = 360
xmin = 0
xaxis = "place=tbci line=1 mark=73 font=screen.r.14 from=0. to=360. by=10."
ytitle = Latitude
yfont = screen.b.14r
yscale = s
ynum = 6336
yorg = 0
ymax = -72
ymin = 72
yaxis = "place=tbci line=1 mark=36 font=screen.r.14 from=-80. to=80. by=10."
FINIS=
```

6) GeoTIFF - Digital Elavation Model

The DMA stores digital elevation models using an equirectangular projection, so that it may be indexed with WGS84 geographic coordinates. Since elevation postings are point-values, the pixels should not be considered as filling areas, but as point-values at grid vertices. To accommodate the base elevation of the Angeles Crest forest, the pixel value of 0 corresponds to an elevation of 1000 meters relative to WGS84 reference ellipsoid. The upper left corner is at 120 degrees West, 32 degrees North, and has a pixel scale of 0.2 degrees/pixel longitude, 0.1 degrees/pixel latitude.

```
      ModelTiepointTag=(0.0, 0.0, 0.0,  -120.0,       32.0,    1000.0)
      ModelPixelScale = (0.2, 0.1, 1.0)
      GeoKeyDirectoryTag:
            GTModelTypeGeoKey          =  2     (ModelTypeGeographic)
            GTRasterTypeGeoKey         =  2     (RasterPixelIsPoint)
            GeographicTypeGeoKey       =  4326  (GCS_WGS_84)
            VerticalCSTypeGeoKey       =  5030  (VertCS_WGS_84_ellipsoid)
            VerticalCitationGeoKey     =  "WGS 84 Ellipsoid"
            VerticalUnitsGeoKey        =  9001     (Linear_Meter)

   Remarks:
           1) Note the "RasterPixelIsPoint" raster space, indicating that
              the DEM posting of the first pixel is at the raster point
              (0,0,0), and therefore corresponds to 120W,32N exactly.
           2) The third value of the "PixelScale" is 1.0 to indicate
              that a single pixel-value unit corresponds to 1 meter,
              and the last tiepoint value indicates that base value
              zero indicates 1000m above the reference surface.
```

7) KML – image tile to place on the earth

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.1">
<Document>
        <name>Topography V12.1</name>
        <GroundOverlay>
                <name>topoN20W160</name>
                <Icon>
                        <href>http://topex.ucsd.edu/kml/topo/R3/topoN20W160.png</href>
                </Icon>
                <altitudeMode>relativeToSeaFloor</altitudeMode>
                <altitude>0</altitude>
                <LatLonBox>
                        <north>20.000000</north>
                        <south>15.000000</south>
                        <east>-155.000000</east>
                        <west>-160.000000</west>
                </LatLonBox>
                <Region>
                <LatLonAltBox>
                        <north>20.000000</north>
                        <south>15.000000</south>
                        <east>-155.000000</east>
                        <west>-160.000000</west>
                </LatLonAltBox>
                <Lod>
                        <minLodPixels>192</minLodPixels>
                        <maxLodPixels>-1</maxLodPixels>
                        <minFadeExtent>40</minFadeExtent>
                        <maxFadeExtent>40</maxFadeExtent>
                </Lod>
                </Region>
        </GroundOverlay>
</Document>
</kml>
```