Mosaicking Distribution List (15 Total)

Name	M/S	Ext
Elaine Chapin	300-235	4-0497
Anhua Chu	300-243	4-0496
Eric J. Fielding	300-233	3-3614
Adam P. Freedman	300-235	4-9072
Scott Hensley	300-235	4-3322
John Holt	300-227	4-6341
Ian R. Joughin	300-235	4-1587
Yunjin Kim	300-243	4-9500
Jan M. Martin	300-235	4-0025
Francois Rogez	300-233	
Paul A. Rosen	300-235	4-0023
Eric Rignot	300-235	4-1640
Scott Shaffer	300-235	4-1217
Mark Simons	252-21 Caltech	395-6984
Paul R. Siqueira	300-227	3-9285
Jakob van Zyl	300-227	4-1365

JET PROPULSION LABORATORY TECHNICAL MEMORANDUM

334-97-004
January 30, 2000

TO Jakob van Zyl
FROM Elaine Chapin
SUBJECT Users Guide for Multimosaic: Version 1.1

1 Abstract

To assist with training the Calgis representative in interferometric processing as part of the GeoSAR project, users guides have been written for the IFPROC interferometric processor, for the mosaicking software package, and for the multimosaic program. This document is the Users Guide for the Multimosaic program. Postscript and HTML versions of this document should be available shortly at `http://www-radar.jpl.nasa.gov/geosar/training/`

2 Introduction

The mosaicker written by Scott Shaffer and Scott Hensley is a useful tool when working with SAR data. Although primarily designed to mosaic together multiple data sets into a single image, the software can perform many day to day tasks, such as rotating data so that two data sets are in the same coordinates and converting data from one projection to another. It is extremely easy to use. The only thing preventing wider use of this software is that the documentation for it has not been written yet. This document is attempts to fill that need; it is an incomplete operational set of instructions and hints originally intended to help in interferometric processor training as part of GeoSAR. This document reflects my involvement with this software as one of the "beta testers". These comments apply only to running on `argus.jpl.nasa.gov`. [1] I will try to add things as appropriate. If you find any errors or have suggestions, please tell me.

The term "mosaicker" is ambiguous since it can refer either to the `multimosaic` program or to the full suite of mosaicking software. Executables of the mosaicking software are in the `argus` directory `/users/topsar/Util2`. Software in the mosaicker suite of code include:

- `ampflat`
- `multimatch`
- `multicull`
- `multiaffine3d`
- `multimosaic`
- `multitps`

[1] Executable versions of every program mentioned in this document exist on `sardine.jpl.nasa.gov` in directory `/export/air1/airops/argus_ifproc/`. The executables in this directory are automatically recopied weekly from `argus` to insure that `sardine` always has up-to-date code.

If all you want to do is to dead reckon a group of files together, then the only routine that you will need is `multimosaic`. That is the only routine I will discuss here. A companion memo is being written to describe the other components of the package. Scott Shaffer and Scott Hensley are in the process of writing a New Technology Report and a journal article describing the mosaicking software.

The mosaicking software continues to evolve, far more quickly then the documentation. Feature that are planned to be implemented soon are included here and marked NYI (Not Yet Implemented).

3 General Format of the Mosaicker Command File

The general format of the mosaicker command files is given below:

```
output_file_name
output file characteristics

input_file_name_11
input file 1 characteristics

input_file_name_22
input file 2 characteristics
```

Below is a real mosaicker command file that was used to dead reckon two antiparallel Rosamond data scenes into a single mosaic.

```
rmg=mos_out.rmg
spc=5.0,5.0
typ=SCH
peg=34.8,-118.07,170.0
fln=50

mgh=rosa96170-3.mag,rosa96170-3.dte
hdr=rosa96170-3.hdr

mgh=rosa96350-1.mag,rosa96350-1.dte
hdr=rosa96350-1.hdr
```

The first line of the example command file specifies the file name for the output data file, `mos_out.rmg`, and specifies the format that the file is to be written in, `rmg`. The second through fifth lines specify some of the characteristics of the output file. It will be an `SCH` file with 5 m postings and a peg at 34.8° latitude, -118.07° longitude, and a heading of 170.0° from north. Since no size or offset was specified, the mosaicker will calculate the appropriate file size and offset to get all of the data in the two files in the mosaic. The feather length parameter indicates that the seams between the two data sets are to be smoothed over 50 pixels.

Lines 7 and 10 give the names of the two input files to be mosaicked. Lines 8 and 11 give the names of the header files for each input data file. Since the input data files were IFPROC output files, all of the information about the sizes, pegs, etc. for the input files is in the IFPROC produced

header files. Listing the header file below the data input files instructs the mosaicker to read the parameters it needs out of those files. In fact it wasn't need to include those header lines. If the headers are named using the default IFPROC file naming convention (which is height file = root.dte, amplitude file = root.mag, and header file = root.hdr), the mosaicker will find and read the header files automatically.

4 Specifying the File Name

A given data file is a certain type, kind, and format. The type is the projection of the data, for example `UTM` or `SCH`. The format is a description of how numbers are represented in the file, namely either signed 1 byte integer, unsigned 2 byte integer, or 4 byte real. The kind is the physical interpretation of the file. Allowed kinds of files are:

rmg Richard M. Goldstein format where both the amplitude and the heights are packed into the same file, with amplitudes on the first half of each line and heights on the second half of each line. All values must be 4 byte reals. This is the default. To combine separate magnitude and height files or to separate an rmg file into magnitude and height files use the commands `nbymmaghgt` and `separate`, respectively. Note that the rmg format is defined as a headerless 4 byte real combined amplitude and height file with no scaling or offset. Several mosaicker options described later such as `dss, mss, mbp`, and `dbp` have no effect on rmg files because the format is strictly defined.

mgh Separate amplitude and height files.

mag Magnitude only data file.

dte Height only data file.

Note that the `mgh` tag is not equivalent to a pair of separate `mag` and `dte` tags. `mgh` assumes that the two files in its argument are corresponding magnitude and height files with the same dimensions. The separate tags imply that the two data files are not related to each other. The kinds `d*2`, `dma`, `m*2` are no now longer needed since the mosaicker can handle integers using the `mbp` and `dbp` fields. Similarly, the kinds `pwr` and `dtm` are being phased out since they are identical to `mag` and `dte`, respectively.

5 File Characteristic Arguments

In addition to the kind and name of the data file, you also need to specify the characteristics of the data file. Below is a list of the possible characteristics for input and output files.

`typ` This argument gives the type of the file. The default value is `SCH` (along track, across track, and height). Other possible arguments are `EQA` (equiangular), `UTM` (Universal Transverse Mercator), `NEU` (northing, easting, and up), `ENU` (easting, northing, and up), `XYZ` (XYZ), `TCN` (tangential, cross, normal), and `SRH` (right looking SCH). This argument is also known as `prj`, projection.

`spc` This argument gives the pixel spacing. For all file types except `EQU` this is a value in meters. For `EQU` files, this is in degrees. The default is `spc=1.0,1.0`. For input files, this argument should be specified either in the command file or in the header file. For more information concerning the order and sign of the `spc` command, see Section 7.

`hdr` This argument gives the name of the header file corresponding to the data file. The header must be in the format used by the IFPROC processor. (NB The AIRSAR processor embeds the header information into the data file. The mosaicker deals with this completely transparently to the user, reading the embedded header and recognizing that the header isn't actually data.)

`peg` This argument specifies the latitude, longitude, and the heading for `SCH`, `TCN`, and `SRH` files. These angles are specified in degrees, not radians. If the file type is `UTM`, the peg's latitude and longitude are used to determine the zone, if the zone is not specified. For other file types this argument is not used.

`zon` This argument is the `UTM` zone. If the file is not a `UTM`, `ENU`, or `NEU` type file, this argument is not used. The alternative is to specify a peg instead. The earth is divided into 60 zones each generally 6° wide. The zones are numbered from 1 to 60 proceeding east from the 180th meridian from Greenwich.

`aff` This specifies the affine transformation file to be applied to the data. The default is that no affine transformation is applied.

`siz` This specifies the number of pixels in the file. For an output file, if no value is given or the value given is equal to zero, then the mosaicker will calculate a size sufficient to hold all of the input data. The output file computation is done independently for each direction. If an output file size is negative, then the absolute value of the size is used. For more information concerning the order of the `siz` command, see Section 7.

`off` This specifies the offset for the data file. This is the position which a (0,0) (or (-1,-1) for you C users) element of the data file would have. The offset is given in the same units as the spacing, `spc`. The order for the two offsets is the same as for the spacing. For output files if the size is less than or equal to zero or no size was given, then the mosaicker will calculate an appropriate offset. If in addition a negative value is given in the output size field, the command line offset field value will be added to the calculated appropriate offset. For more information concerning the order of the `off` command, see Section 7.

`wgt` This is the weighting factor. The default is to weight all of the input files' data by 1.0 when mosaicking the data together. This parameter can be used to adjust the relative weighting of the input files, using a small number for poorer quality data and a larger weighting factor for higher quality data. Note that if you weight one data set very heavily relative to the other, feathering will not be very effective.

`dto` This is the number of lines to skip from the front of a height image before the first image data. Useful for data files with headers prefixed directly into the file. The default value of this parameter is 0. See section 8.

`mgo` The number of lines to skip from the front of a magnitude image. Similar to `dto`. See section 8.

`mhl` Magnitude File Header Lines

`mhs` Magnitude File Header Samples eg. The number of bytes skipped for a 2 byte integer magnitude data file would be mhl*mhs*2, and the number of bytes skipped for a 4 byte real height file would be dhl*dhs*4.

`dhl` Elevation File Header Lines

`dhs` Elevation File Header Samples

`dss` The scale and shift for the height file is given by `dss`. This is normally used with 2 byte integer height data, which is often shifted by a 100 meters to make all heights positive and represented in centimeters instead of meters. To represent the example situation described you would use the command `dss=0.010,-100.00`. The formula used is
$value_{in\_file} = (1/scale) * (value_{in\_physical\_units} - shift)$.
scale is sometimes referred to as the quantization step size, and $shift$ is sometimes called the minimum value. Note that 2 byte integers in the mosaicker are unsigned.

`mss` Same as `dss` except applies to the magnitude file rather than the height file.

`mbp` This is the number of bytes per pixel for the magnitude file and must be either 1 (1 byte integers), 2 (2 byte unsigned integers), or 4 (4 byte reals). The default is 4. At this time, the mosaicker does not handle complex data or 8 byte real data.

`dbp` This is the number of bytes per pixel for the height file. Allowed values are either 1 (1 byte integers), 2 (2 byte unsigned integers), or 4 (4 byte reals). The default is 4.

`did` The datum identifier is used to indicate the datum for a data set. Possible values include `WGS84`, which is also the default, and `NAD27`. (NYI)

`mmx` The default is that all non-zero input magnitude values correspond to valid data. If set, all input magnitude values greater than `mmx` will be considered invalid. If the `mmx` option is used with an output file, all output values greater then `mmx` will be written to the file as `mmx`. `mmx, mmn, dmx, dmn, min`, and `max` are all in the real scaled and shifted system. For example, if the `mmx` value is set at 3276.8 m and the output is 2 byte integers shifted and scaled by 0.0 and 0.1, then data values greater then 3276.8 will be written to the output file as the integer 32768. (NYI)

`mmn` The default is that all non-zero input magnitude values correspond to valid data. If set, all input magnitude values less than `mmn` will be considered invalid. If the `mmx` option is used with an output file, all output values less then `mmn` will be written to the file as `mmn`. Similar to `mmx`, only `mmn` sets the minimum value not the maximum value. (NYI)

`dmx` This characteristic is similar to `mmx` except applies to height data files. If set, all height values greater than `dmx` will be considered invalid. (NYI)

`dmn` This characteristic is similar to `mmn` except applies to height data files. If set, all height values less than `dmn` will be considered invalid. (NYI)

`rad` The argument to `rad` is the number of terms to use in the polynomial correction to the amplitude. `rad=0` turns off this feature completely. With this feature on, the amplitude of the pixels as a function of the local slope are fit with a polynomial of degree (`rad`-1) and then the fitted curve is removed from the amplitude data. This function is similar to `ampflat` and `ampfcheby`, but attempts to also correct for the local slope. (NYI)

6 Mosaic Characteristics

In addition to setting characteristics of the individual input and output files, there are characteristics to set that apply to all of the files. Below is a list of these characteristics.

`min` The mosaicker assumes that any data point with an amplitude of zero or a height outside of a user setable range is not valid data. The default height range is between -500 and 9000 m. Note that the range is specified in terms of the scaled and shifted data values, not in terms

of the values read directly out of the data file. `min` is used to adjust the data limit from the default of -500 m to `min`'s argument. This characteristic differs from `mmn` in that it applies globally to all input files.

`max` `max` is used to adjust the maximum valid height data limit from the default of 9000 m to `max`'s argument. This characteristic differs from `mmx` in that it applies globally to all input files.

`nul` `nul` gives the null value for height data. By null value, we mean the value in the height file corresponding to missing heights. Normally this value is -10000.0.

`gcp` This is the name of the ground control points file. The mosaicker expects that this file will have three ascii columns: the latitude in decimal degrees, the longitude in decimal degrees, and ellipsoid height in meters. The time is not used. The position of the ground control points are marked on the output mosaic using diamonds of brightness given by `gfv` and a size given by `gfs`. The height at the pixel location closest to the ground control point is replaced by the ground control point value in the outputted DEM. The `gcp` keyword mimics the functionality of programs such as `makemapgps`.

`gfs` This is the fill size for the ground control points marked on the output mosaic given in pixels. Larger values of `gfs` correspond to larger diamonds marking the position of the points on the output mosaic. The width of the diamond is `gfs`*2-1 pixels.

`gfv` The value to fill in the amplitude set at the postion of ground control points. The default is 10000.

`fln` This is the feathering length. The maximum value is 50 pixels. This parameter defines how the data from each file is weighted in overlap regions near seams. This characteristic can be specified anywhere in the command file but can only be applied to the output file.

`std` This parameter is the name of the standard deviation map. If this argument is not present, no standard deviation map will be made. The standard deviation map outputs the standard deviation of the height estimates at each pixel when overlapping data is present. When there is no overlap, the standard deviation is zero.

`smp` This is the sampling method. Valid values are `nearest` neighbor and `simplicial`. The default is `simplicial`.

`cmb` Possible values for this amplitude combination method flag are `max`, `min`, and `avg`, where `avg` is the default. `avg` means that the of the amplitude data from the various input files for a given pixel will be averaged to produce the output mosaicked magnitude image. For `min` and `max`, the minimum or maximum of all the the input data will be selected for each pixel. `min` and `max` disable any weighting assigned to that input data.

`fac` The look down factor is given by this argument. This does not give the look down factor of the input file relative to the header the way that `dgx`'s `f` option does. Instead this tells the mosaicker how many lines and samples to skip when reading the input files. For example, suppose that the you have two 5 meter posted input files, but you are requesting a mosaic with a 100 meter posting. If you ran with `fac=10` the mosaicker would run much faster because it would only read every 10th line. The output would look almost the same since the input samples would still be twice as frequent as the output pixels. The mosaicker actually calculates and uses a conservative value of this parameter. In the sample mosaicker output lines:

```
at block            2           6
  Reading file            1            1160        2002           1
  Reading file            2             563        1525           1
  Writing block
```

the 1's in the fourth column indicates that a `fac` of 1 was used for input files. This parameter is useful when mosaicking large high resolution files to low resolution. It is useful because the mosaicker does not interpolate or multilook the input data. A `fac` field given in the characteristics for a given input data set applies only to that set. If a `fac` field is given in the output file characteristics section of the command file, it applies to all of the input data except for sets for which a `fac` field is given in the set's own characteristics field.

7 The Order and Sign for Spacing, Size, and Offset

One of the frequent areas of difficulty is order and sign for the characteristics `spc`, `siz`, and `off`. For most file types, the mosaicker assumes that the data in the file is arranged in the following order:

$$\begin{array}{ccccc} (U_1, V_1) & (U_1, V_1 + s_V) & (U_1, V_1 + 2s_V) & \dots & (U_1, V_1 + N_V s_V) \\ (U_1 + s_U, V_1) & (U_1 + s_U, V_1 + s_V) & (U_1 + s_U, V_1 + 2s_V) & \dots & (U_1 + s_U, V_1 + N_V s_V) \\ (U_1 + 2s_U, V_1) & (U_1 + 2s_U, V_1 + s_V) & (U_1 + 2s_U, V_1 + 2s_V) & \dots & (U_1 + 2s_U, V_1 + N_V s_V) \\ \dots & & & & \\ (U_1 + N_U s_U, V_1) & (U_1 + N_U s_U, V_1 + s_V) & (U_1 + N_U s_U, V_1 + 2s_V) & \dots & (U_1 + N_U s_U, V_1 + N_V s_V) \end{array} \quad (1)$$

where `off=`U_0, V_0 (the location one unit from the first data point in the file), `spc=`s_U, s_V (the spacings between data points), `siz=`N_U, N_V (the number of samples and lines in the data file), (U_1, V_1) and $(U_1, V_1 + s_V)$ are adjacent in memory, and (U_1, V_1) and $(U_1, V_1 + 2s_V)$ are separated in memory by N_V pixels. For the different file types, U and V are defined as follows:

Type	U	V
`SCH`	Along Track Distance (m)	Across Track Distance (m)
`NEU`	Northing (m)	Easting (m)
`ENU`	Easting (m)	Northing (m)
`EQU`	Latitude (degrees)	Longitude (degrees)
`XYZ`	X (m)	Y (m)
`TCN`	Transverse Distance (m)	Cross Track Distance (m)

The `SRH` and `UTM` file types do not fit the pattern outlined above. `UTM` is identical to `NEU` except the positive direction for the spacing is defined to be south. `SRH` is identical to `SCH` except the across track spacing and offset are right looking not left looking.

For example, Bob Crippen's mosaic of the USGS's 90 m data, `/data/par/CA-NEV-DEM/R.Crippen.Mosaic`, is an equiangular file. The data is on a uniform grid with a 8.33333e-4° spacing, running from -125.0 to -114.0° longitude and from 32.0 to 42.0° latitude. The first point in the data file is at (42.0°,-125.0°), the northernmost (largest latitude) and the westernmost (smallest longitude) point in the set. Hence, for this datafile `off=42.0,-125.0`, `spc=-8.33333e-4,8.33333e-4`, and `siz=12000,13200`. Note that the latitude spacing is negative.

The Long Valley mosaic is an example of a UTM formatted file. The first element in the data file is at (37.8957°,-119.3125°) which corresponds to a northing and easting of 4196780.00 and 296665.0, respectively. This point is the northernmost and westernmost. The last data point in the file is approximately 38 km south and 117 km east of the first data point. Hence, for this datafile `off=4196780.00,296665.00`, `spc=5.0,5.0`, and `siz=8075,23397`. Note that if this file's type had been given as NEU the offset, spacing, and size would be `off=4196780.00,296665.00`, `spc=-5.0,5.0`, and `siz=8075,23397`, respectively. Note that the northing spacing is negative.

8 Embedded Headers

This section discusses the proper use of `mhl, mhs, dhl, dhs, mgo`, and `dto`.

For files with embedded headers, the `siz` specifier gives the number of image pixels, not the file dimensions. For example if in the command file you specified for one of the input files that:

```
siz=1000,1200
mhl=1
mhs=16
mbs=4
```

then the number of bytes per line of magnitude file is 4*(1000+16) = 4064, the number of bytes per line of the magnitude file that is image is 4*(1000) = 4000, the total file size in bytes is 4064*(1200+1) = 4880864, and the total image size in bytes is 4000*(1200) = 4800000.

The default value of `mhl, mhs, dhl, dhs, mgo`, and `dto` is 0.

9 Precedence Rules

If the mosaicker is given contradictory information about a data file in the data file's internal header, external header, and the mosaicker command file, which one does the mosaicker use?

1. Header files internal to the data file are highest priority.
2. The mosaicker command file is the second highest priority.
3. The external to the data file separate header file is third priority.
4. For the output file only, information about the peg, spacing, zone (NYI), type (NYI), and datum identification (NYI), are taken from the first input image if not specified in the mosaicker command file or the external header for the output data file. (The mosaicker doesn't output data with an internal header file.) Note that the mosaicker will not read the output file size or the output file offset from the first input image.

10 Other Tips on Constructing a Mosaicker Command File

There are some other aspects of making a mosaicker command file that require your attention.

- It is conventional, but no longer required, to leave one hard carriage returned blank line at the end of the command file, to leave one blank line between information about each file, and not to leave any spaces within a line.
- You are only allowed 200 input files.
- Comments are made by putting a # sign at the front of the line that you want to comment out.
- Do not leave any blank lines at the beginning of the input file.
- All lengths are in meters. All angles are in degrees. The only fields given in pixels are `siz` and `fln`.

- The default file type is `SCH`. Any other type must be specified. If all of the input files were, for example, `UTM` and no output file type was specified, the output file would be `SCH`, not `UTM`.
- In the files output by the mosaicker, there may be some pixel locations which were not included in any on the input images. For those pixels, the amplitudes and heights will be set to zero. For some pixels, there may be input height data but no input amplitude data. For those pixels the amplitudes are set to one. For the cases where there is amplitude data but no height input data, the pixels' heights are set to zero.
- The file charactaristics and mosaic charactaristics were given in Sections 4, 5, and 6. For each input file, you must tell the mosaicker either via the mosaicker command file or the input file header: what the file is (rmg, mgh, mag, dte, hdr), how the data is formatted in the file (dto, mgo, dss, mss, min, max, mmx, mmn, dmx, dmn) where the file is geographically (typ, zon, peg, off, aff, did), and how big the file is (mbp, dbp, spc, siz). Generally, not all of these will need to be specfied. In fact, for a given input data set one can only specify rmg or mgh or mag or dte, specify max or dmx, and specify min or dmn.
- When outputting one or two byte files, data that exceeds the maximum and minimum value is clipped, not wrapped. For example, if the data goes from -100.0 to 10000.0 m, you are outputting two byte heights, and you specified a `dss` of 0.1, 0.0, then all the data from -100.0 to 0.0 m will be output to integer value 0 in the output file and all the data from 6553.6 to 10000.0 m will be output to the integer value 65536.
- It is strongly recommended that you never use eighth bit of one byte data or the sixteenth bit of two byte data and that you make all outputs posative. Some software expects data with sign bits and some without sign bits. Using one fewer bit greatly reduces confusion for a very small price.

11 Examples of Mosaicking Command Files

Mosaic AIRSAR data into UTM The example below creates a UTM file from the SCH AIRSAR magnitude and height files. On output we are specifying 10 m postings. The peg call is used to set the zone. For this example the magnitude file is in the AIRSAR real*4 format and the height file is in the i*2 format. The alternative choices are commented out. No header is needed since the header info is embedded in the data in the AIRSAR format.

```
mgh=mosaic_utm.vvgr,mosaic_utm.demgr
spc=10.0,10.0
typ=UTM
peg=34.8,-118.07,170.0

mgh=airsar.vvgr,airsar.demi2

# mgh=airsar.vvi2,airsar.demgr
```

Mosaic IFPROC Data into Coordinates of Another IFPROC Run Let us suppose that I have two runs that I have processed and that I wish to compare the heights in the overlapping region. The mosaicker, using the command file below will mosaic the data from the Rosamond 350-1 run into a file with the same peg and upper left corner position as the already processed Rosamond 170-3 run. The command file uses the header file from the Rosamond 170-3 run to set the output file specifications. After mosaicking, I could easily difference the two rmg files using `diffhgt`.

```
rmg=rosa96350-1_170-3.rmg
hdr=rosa96170-3.hdr

mgh=rosa96350-1.mag,rosa96350-1.dte
```

Mosaic IFPROC Data into Separated Two Byte Integer Files Let us suppose that I have processed a 1996 Rosamond 350° heading run and would like to convert it to two byte integers. The scale and shift parameters have been specifically chosen to fit the data set. For this set, the height data varies from 610 to 1219 m, and the magnitude data's mean, standard deviation, and minimum are 0.389, 0.402, and 0.000, respectively. The Table below shows how some representative values in the original data will map into the unsigned 2 byte integers.

v_{in_file}	$v_{in\_physical\_units}$	$scale$	$shift$
0	600.000	0.05	600.0
200	610.000	0.05	600.0
201	610.050	0.05	600.0
12380	1219.000	0.05	600.0
2^{15}	2238.400	0.05	600.0
0	0.00	0.000061	0.0
6377	0.389	0.000061	0.0
2^{15}	1.998	0.000061	0.0
2^{16}	3.998	0.000061	0.0

Using the 5 cm step size and the 600.0 m shift, allows all of the height data to easily map into only 15 of the 16 bits in the unsigned 2 byte integers. An offset of 0 was chosen for the amplitude data because that is the minimum for the data. By choosing a step size of 0.000061 for the amplitude data, data values more then 9 standard deviations greater than the mean will be truncated at that value. ($\mu = 0.389$, $\sigma = 0.402$,$\mu + 4\sigma = 1.998$, $\mu + 9\sigma = 3.998$) This means that the full 16 bits of the unsigned 2 byte integers may be exercised.

```
mgh=rosa96350-1i2.mag,rosa96350-1i2.dte
mbp=2
mss=0.000061,0.0
dbp=2
dss=0.05,600.0
peg=34.8382798,-117.9877259,-9.9637942
spc=5.0,5.0

mgh=rosa96350-1.mag,rosa96350-1.dte
```

Mosaic IFPROC Data Together with USGS 90 meter Data Let us suppose that I have two runs that I have processed and that I wish to mosaic them together, filling any gaps with the USGS 90 m data. Note that I will have to specify a size since the the USGS data file contains data from all of California and the mosaicker would set the size and offsets sufficient to include all of the data in all three input files.

```
rmg=rosamerge.rmg
siz=12560,3163
spc=5.0,5.0
off=-30875.0,-7565.0
```

```
peg=34.80,-118.07,170.0

mgh=rosa96350-1.mag,rosa96350-1.dte

mgh=rosa96170-3.mag,rosa96170-3.dte

dte=/data/par/CA-NEV-DEM/R.Crippen.Mosaic
hdr=/data/par/CA-NEV-DEM/R.Crippen.hdr
wgt=0.001
```

12 Running the Mosaicker

Writing the command file is the most difficult part of running the mosaicker. To run a command file `file.cmd` the format would be
`multimosaic file.cmd`.
The mosaicker will then write to the screen the name of the command file that you typed on the command line and a list of all of the input files that it will read. Then it will list the name of the output data and header files that it will create. If a header file of the same name already exists, it will confirm that you wish to overwrite the file. Next it will write information about each block within the mosaic as it is written. The program gives you the cputime used as it terminates. Below is a sample screen output.

```
<< Multimosaic   Version 1.0 >>


 Opening cmd input  file: mosaic1.cmd


  Opening hdr input  file: rosa96170-3.hdr
  Opening amp input  file: rosa96170-3.mag                          2560 11485
  Opening hgt input  file: rosa96170-3.dte                          2560 11485

  Opening hdr input  file: rosa96350-1.hdr
  Opening amp input  file: rosa96350-1.mag                          2560 11523
  Opening hgt input  file: rosa96350-1.dte                          2560 11523

  Opening rmg output file: mosaic1_output.rmg                       1213  1236

  Writing hdr output file: mosaic1_output.hdr

  Output file offsets =    13761900.00000000       5101600.000000000

  at block            1           1
   Reading file            1             2560       11485           3
   Reading file            2             2560       11523           3
   Writing block

  Total run time
```

```
     cputime:    289.4115
     systime:    9.207222
STOP Done statement executed
```

13 Appendix: Notes on Constructing Header Files

Below is a typical header file using by the mosaicking software.

```
                                SCH ;  Data file type
10000   2560                        ;  Data file dimensions
5.000  5.000                        ;  Post Spacing
-14885.00 4265.00                   ;  Starting corner position (s,c)
-43.4561289 170.3659660 135.0661715 ;  Peg position (WGS-84)
```

The basic format in the file is values semicolon keyword. This format is not a valid RDF format. Below is a list of valid keywords to include in the header. Case, spelling, and punctuation of keywords are important.

Command File Charactaristic	Header File Keyword
`typ`	Data file type
`siz`	Data file dimensions
`spc`	Post Spacing
`off`	Starting corner position (s,c)
`peg`	Peg position (WGS-84)
`zon`	Zone
`mss`	Magnitude Scale and Shift
`dss`	Elevation Scale and Shift
`mbp`	Magnitude Bytes per pixel
`dbp`	Elevation Bytes per Pixel
`aff`	M11 M12 M13 M21 M22 M23 M31 M32 M33 O1 O2 O3
`mhl`	Magnitude File Header Lines (mhl)
`mhs`	Magnitude File Header Samples (mhs)
`dhl`	Elevation File Header Lines (dhl)
`dhs`	Elevation File Header Samples (dhs)