

Airborne Processing Library

Advanced APL - Command line

ARSF - Data Analysis Node

August 13, 2012

Contents

1	Advanced APL - Command line	2
1.1	Files required for tutorial: Advanced APL - Command line . . .	2
2	Data description	2
2.1	Level 1 imagery	2
2.2	Level 1 navigation	3
2.3	Digital Elevation Model	3
2.4	Sensor view vectors	3
3	Procedure description	3
4	Command line procedure	4
4.1	Generate per-pixel items suitable for atmospheric correction algorithms	4
4.2	Creating an IGM without a Digital Elevation Model	5
4.3	Re-projecting data into a user defined projection	5
4.4	Resampling to level-3 using different interpolation methods	6
4.5	Mapping a subsection of a flight line	7
4.6	Selecting different groups of bands to map	8
4.7	Ignoring certain data	9
5	Review	10
A	Appendix - File list	11
B	Appendix - Installation	12

1 Advanced APL - Command line

This tutorial follows up on previous ones by introducing some more advanced options of APL. By the end of it you should be able to produce files containing per-pixel view vector information, Digital Elevation Model slope and aspect, use user specified re-projections, use different interpolation algorithms and map specific areas of a flight line. Some of this information may be important if you wish to use 3rd party atmospheric correction algorithms (these are not currently part of APL).

Example data is included and can be downloaded from the ARSF Data Analysis Node website. Note that running any of the APL programs with “-help” will give brief notes about all the available options.

Note also that commands that are designed to be executed are **highlighted**. It is envisaged that the tutorial will take approximately 30 minutes to complete.

1.1 Files required for tutorial: Advanced APL - Command line

This tutorial requires the file `apl_basic_tutorial_data.zip` to be downloaded from the ARSF-DAN website from:

http://arsf-dan.nerc.ac.uk/files/apl_basic_tutorial_data.zip .

The contents of this file can be found in the appendix in section [A](#).

2 Data description

The example data for this tutorial are in ENVI file formats. ENVI is a commercial image processing package. The data format is very simple and can be read by numerous software including the open source GDAL libraries. The data formats used are BIL (band interleaved by line) and BSQ (band sequential). These are binary file formats with an accompanying text header file that contains meta-data file descriptions.

The example data is contained in an archive structured similarly to how ARSF deliver the hyperspectral data. The layout is as follows.

2.1 Level 1 imagery

The level 1 image data is found in the `flightlines/level1b` directory. Each flightline has 2 files; one ending with ‘.bil’ and one with ‘.hdr’ as described above. These contain radiometrically calibrated “at sensor radiance” values.

2.2 Level 1 navigation

The navigation data is found in the flightlines/navigation directory. Each flightline has 2 files. These contain the time, position and attitude of the aircraft (including sensor boresight and lever arm information) synchronised to the level 1 image scan lines.

2.3 Digital Elevation Model

A Digital Elevation Model (DEM) is included with the example data in the dem directory. This has been created from ASTER data and covers the area of the included flight lines. The DEM is in APL format - which is a BSQ file with the data in WGS84 Geographic latitude/longitude with heights referenced to the WGS84 spheroid.

2.4 Sensor view vectors

The sensor view vector files are in the sensor_FOV_vectors directory. These are BIL files that contain the sensor CCD view vectors for the full CCD. Note that these may not correspond directly to the level 1 image data as the Eagle instrument CCD also captures data from the FODIS sensor.

3 Procedure description

This tutorial is designed to get users more comfortable using the advanced features of APL on the command line. By the end of it we shall have mapped a specific area of interest of a flight line, created new output files containing per-pixel view vector information and introduced some new APL options. More specifically:

- Use `aplcrr` to generate per-pixel azimuth/zenith view vector data and DEM slope/aspect data
- Use `apltran` to reproject data into user specified projections
- Use `aplmap` to resample the level 1 data into a regular grid using different interpolation algorithms
- Use `aplmap` to only map a specific region of the flight line.

4 Command line procedure

The following section gives a step-by-step procedure for processing the example data using the command line. It is assumed that you have downloaded APL prior to this, if not please do so now (see section B). It also assumes a basic working knowledge of using APL as gained from the basic APL tutorials.

4.1 Generate per-pixel items suitable for atmospheric correction algorithms

Here we will use `aplcrr` to generate an additional file that contains the following items in the form of a BIL file:

- Sensor to ground vector in terms of distance, azimuth and zenith angles
- Digital Elevation Model slope and aspect angles

From the basic tutorial we learnt that a suitable command line to process the `level1` data to get an IGM file is:

```
aplcrr -lev1file flightlines/level1b/eagle_line.bil \  
-navfile flightlines/navigation/eagle_line_nav_post_processed.bil \  
-vvfile sensor_FOV_vectors/eagle_fov_fullccd_vectors.bil \  
-dem dem/test-site.dem \  
-igmfile outputs/eagle_line.igm
```

To output the additional file we need to use the `'-atmosfile'` argument on the command line, passing it a name to give to the new file:

```
aplcrr -lev1file flightlines/level1b/eagle_line.bil \  
-navfile flightlines/navigation/eagle_line_nav_post_processed.bil \  
-vvfile sensor_FOV_vectors/eagle_fov_fullccd_vectors.bil \  
-dem dem/test-site.dem \  
-igmfile outputs/eagle_line.igm \  
-atmosfile outputs/eagle_line_extraparameters.bil
```

Running the above command will result in an IGM file being created as usual, but with the extra file `line1_extraparameters.bil` also being created. This is a 5-band BIL file where each band has per level-1 pixel values ordered such that:

- Band 1: view azimuth: angle in degrees, clockwise from North, for the sensor to ground intersect

- Band 2: view zenith: angle in degrees, measured from 'up', for the sensor to ground intersect
- Band 3: distance: distance in metres from sensor to ground intersect
- Band 4: dem slope: angle in degrees measured from horizontal
- Band 5: dem aspect: angle in degrees, clockwise from North

4.2 Creating an IGM without a Digital Elevation Model

When generating an IGM file you will **always** get better results using the most accurate DEM. But occasionally it may be that there is not a DEM available for the required area, or - in the case of mapping ocean areas - using a DEM is not appropriate. In these rare cases it is possible to generate an IGM.

In the simplest form, an IGM can be created by just not including the '-dem' option on the command line. In this case aplcorr will use the WGS84 spheroid surface to project on to. This will contain obvious errors unless the area being mapped is smooth and coincident with the WGS84 surface.

If an estimate of the surface height is known then it is possible to use the '-heightoffset' option, which will tell aplcorr to use the a spheroid surface of a certain height above the WGS84 spheroid. This can be useful for when mapping data acquired over expanses of water, where the surface is assumed smooth and at a constant height - but not necessarily at the elevation of the WGS84 spheroid. For example at 54N, 1E (in the North Sea) the mean sea level is (approx) 44m above the WGS84 spheroid. So if we wanted to map some data from this area we would use `-heightoffset 44` to estimate a smooth surface 44m above the WGS84 spheroid.

4.3 Re-projecting data into a user defined projection

APL has projection 'keywords' for 2 projections: UTM and Ordnance Survey National Grid. But it also has the capability to use any of the projections built into the [PROJ4](#) library. To use this functionality you need to build the projection information into a proj-style projection string and declare this on the apltran command line using the -outprojstr option.

From the basic tutorial we have learnt that we need to specify the input and output IGM filenames, such as:

```
apltran -igm outputs/eagle_line.igm \
-output outputs/eagle_line_lcc.igm
```

Now we need to specify the output projection using the ‘-outprojstr’ option. Lets say we want to map the data to a Lambert Conformal Conic projection, using the WGS84 datum and spheroid together with latitude of origin 49 degrees and longitude of origin -2 degrees. We’ll also use false easting of 400000 and false northing of -100000, a 1st standard parallel of 45 and a 2nd standard parallel of 55. **Please note** that this projection as set up here is not recommended to be used operationally - it is purely an example to show how you would use a defined projection with APL.

This would be defined in PROJ4 as:

```
+proj=lcc +datum=WGS84 +lat_0=49 +lon_0=-2 +x_0=400000 +y_0=-100000
+lat_1=45 +lat_2=55
```

and can be used in apltran in the ‘outprojstr’ option as:

```
apltran -igm outputs/eagle_line.igm \
-output outputs/eagle_line_lcc.igm \
-outprojstr +proj=lcc +datum=WGS84 +lat_0=49 +lon_0=-2 \
+x_0=400000 +y_0=-100000 +lat_1=45 +lat_2=55
```

For the remaining tutorial subjects we will assume the IGM is mapped in Ordnance Survey National Grid projection. Recall from the basic tutorial that this can be performed using the following command:

```
apltran -igm outputs/eagle_line.igm \
-output outputs/eagle_line_osng.igm \
-outproj osng ostn02/OSTN02_NTV2.gsb
```

where the file ‘OSTN02_NTV2.gsb’ is downloaded from: http://www.ordnancesurvey.co.uk/oswebsite/gps/osnetfreeservices/furtherinfo/ostn02_ntv2.html

4.4 Resampling to level-3 using different interpolation methods

By default aplmap uses nearest neighbour when gridding the image data. It is possible to use alternative interpolation methods if desired. This is done using the ‘-interpolation’ option. As of version 2.4 of APL there are 4 supported interpolation methods:

- Nearest neighbour: Default method that inserts the level-1 data value that falls nearest to the centre of the level-3 grid.
- Bilinear: Uses a bilinear function and the 4 nearest pixels (one from each neighbouring quadrant) to get an interpolated value.

- Inverse distance weighted: Uses the distance of the nearest N points to calculate a weighted value for the level-3 pixel.
- Cubic: Uses a series of 1 dimensional cubic splines and 16 surrounding points (4 from each quadrant) to get an interpolated value.

Nearest neighbour is the fastest method and is guaranteed to only give values that are present in the level-1 data. Bilinear, inverse distance weighted and cubic are all slower but give “smoother” less grainy looking images. Each method has a keyword which is passed to `aplmap` using the ‘-interpolation’ option. From the basic tutorial we have learnt that we can map an image using a command like:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \
-igm outputs/eagle_line_osng.igm \
-mapname outputs/eagle_line_osng_mapped.bil \
-pixelsize 2 2 \
-bandlist 3 2 1
```

To use bilinear interpolation instead of the default nearest neighbour we can use:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \
-igm outputs/eagle_line_osng.igm \
-mapname outputs/eagle_line_osng_mapped.bil \
-pixelsize 2 2 \
-bandlist 3 2 1 -interpolation bilinear
```

Or to use the inverse distance weighted algorithm, with 10 points, we can use:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \
-igm outputs/eagle_line_osng.igm \
-mapname outputs/eagle_line_osng_mapped.bil \
-pixelsize 2 2 \
-bandlist 3 2 1 -interpolation idw 10
```

4.5 Mapping a subsection of a flight line

It may be that you are only interested in a small subsection of a flight line and do not want to map the full line. If the area of interest is known in coordinates then the option ‘-area’ can be used to define the subset to be mapped. The area must be given in the same coordinate system as the IGM file, so if the IGM file is in Ordnance Survey National Grid projection so must the area coordinates.

The area coordinates are given in the order: minimum X maximum X minimum Y maximum Y

Recall that to map the full flight line of data we used the command:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \  
-igm outputs/eagle_line_osng.igm \  
-mapname outputs/eagle_line_osng_mapped.bil \  
-pixelsize 2 2 \  
-bandlist 3 2 1
```

Lets say that we wish to map only the area contained by coordinates 420278 to 421428 East, 218174 to 219136 North. The command can be updated to use the 'area' option thus:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \  
-igm outputs/eagle_line_osng.igm \  
-mapname outputs/eagle_line_osng_mapped.bil \  
-pixelsize 2 2 \  
-bandlist 3 2 1 -area 420278 421428 218174 219136
```

This results in the region in Figure 1 being mapped.

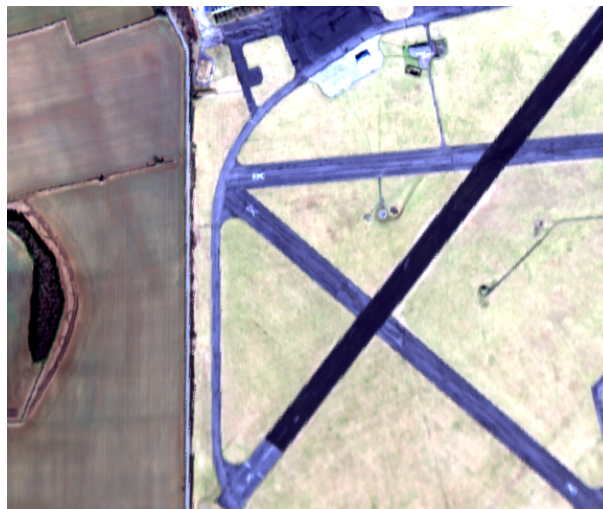


Figure 1: Area of interest mapping with APL.

4.6 Selecting different groups of bands to map

So far we have always selected the bands to map using a space separated string such as: `-bandlist 3 2 1`

It is however possible to use different ways to select groups of bands rather

than creating a long list. If you wish to map all bands of a file then you can use the option as: `-bandlist ALL`

If you wish to map a group of bands, say bands 1 to 25 inclusive, then you can use the option as: `-bandlist 1-25`

If you wish to mix up these methods it is also possible. To map bands 1 to 25, 55, 57, and 124 to 150 the option can be used like: `-bandlist 1-25 55 57 124-150`

This can be tested using the sample Hawk data, as the sample Eagle data only has 4 bands. Recall the 2 commands to create an IGM file in Ordnance Survey National Grid projection:

```
aplcorr -lev1file flightlines/level1b/hawk_line.bil \  
-navfile flightlines/navigation/hawk_line_nav_post_processed.bil \  
-vvfile sensor_FOV_vectors/hawk_fov_fullccd_vectors.bil \  
-dem dem/test-site.dem \  
-igmfile outputs/hawk_line.igm
```

```
apltran -igm outputs/hawk_line.igm \  
-output outputs/hawk_line_osng.igm \  
-outproj osng ostn02/OSTN02_NTv2.gsb
```

Now we can map the data, using the above bands, using the command:

```
aplmap -lev1 flightlines/level1b/hawk_line.bil \  
-igm outputs/hawk_line_osng.igm \  
-mapname outputs/hawk_line_osng_mapped.bil \  
-pixelsize 2 2 \  
-bandlist 1-25 55 57 124-150
```

4.7 Ignoring certain data

It is possible when mapping the data to ignore certain scan lines or data with a certain value. This is useful when you have masked the file and do not wish to map the pixels with the masked value. If you wish to ignore entire scan lines (for all bands) then you can use the option ‘-ignorescan’. So if you wished to ignore scan lines 100, 102 and 103 you could use the option as: `-ignorescan 100 102 103`.

It would be unusual to ignore entire scan lines and more often should like to ignore certain pixels. By default, `aplmap` will ignore data that has a

value of '0'. This corresponds to the value that aplmask will apply to the masked level-1 data. To change this value one can use the '-ignorevalue' option. For example, to ignore data which has the value 1000 you would use: `-ignorevalue 1000`.

If you wish to use all the data in the level-1 files, and not ignore anything, then you can use the keyword 'NONE' as follows: `-ignorevalue NONE`.

Note that the value entered is converted to the same data type as the input level-1 data. For example, the level-1 data included within the tutorial data is unsigned 16-bit integer data. Therefore the values of the data lie between 0 and 65535. If you entered the ignore value as -1 this would be converted (wrapped around) to be 65535. Likewise, if you entered 65536 this would be converted to be 0.

An example of using the 'ignorevalue' option to map all the Eagle level-1 data, that is, not ignoring the masked out dropped scans is:

```
aplmap -lev1 flightlines/level1b/eagle_line.bil \  
-igm outputs/eagle_line_osng.igm \  
-mapname outputs/eagle_line_osng_mapped.bil \  
-pixelsize 2 2 \  
-bandlist 3 2 1 -interpolation bilinear -ignorevalue NONE
```

This can be compared to the earlier image created in section [4.4](#).

5 Review

Advanced command line usage of APL has been introduced for the following points:

- Creating an IGM without a DEM - for use over ocean areas
- Outputting additional information which may be useful for atmospheric correction algorithms
- Re-projecting data using user defined projections
- Resampling using different interpolation methods
- Mapping area of interest of a flight line
- Selecting groups of bands to map
- Change the data value to be ignored when mapping

This concludes the Advanced APL - Command line tutorial.

A Appendix - File list

Table of files contained in `apl_basic_tutorial_data.zip` .

Filename	Description
<code>test-site.dem</code>	Digital Elevation Model made from ASTER data. In WGS84 latitude / longitude and referenced to WGS84 spheroid surface.
<code>test-site.dem.hdr</code>	Header file containing meta-data relating to <code>test-site.dem</code> .
<code>eagle_line.bil</code>	Level 1 processed Eagle imagery
<code>eagle_line.bil.hdr</code>	Header file containing some meta-data relating to <code>eagle_line.bil</code>
<code>eagle_line_mask.bil</code>	Mask file for level 1 Eagle imagery
<code>eagle_line_mask.bil.hdr</code>	Header file containing some meta-data relating to <code>eagle_line_mask.bil</code>
<code>eagle_line_nav_post_processed.bil</code>	Navigation synced to <code>eagle_line.bil</code> data
<code>eagle_line_nav_post_processed.bil.hdr</code>	Header file containing some meta-data relating to <code>eagle_line_nav_post_processed.bil</code>
<code>eagle_fov_fullccd_vectors.bil</code>	Eagle full CCD view vector file. Contains the Eagle CCD view vectors.
<code>eagle_fov_fullccd_vectors.bil.hdr</code>	Header file containing meta-data relating to <code>eagle_fov_fullccd_vectors.bil</code> .
<code>eagle_line.xml</code>	XML meta-data relating to <code>eagle_line.bil</code> .
<code>hawk_line.bil</code>	Level 1 processed Hawk imagery
<code>hawk_line.bil.hdr</code>	Header file containing some meta-data relating to <code>hawk_line.bil</code>
<code>hawk_line_mask.bil</code>	Mask file for level 1 processed Hawk imagery
<code>hawk_line_mask.bil.hdr</code>	Header file containing some meta-data relating to <code>hawk_line_mask.bil</code>
<code>hawk_line_mask-badpixelmethod.bil</code>	CCD detector method mask file for level 1 processed Hawk imagery
<code>hawk_line_mask-badpixelmethod.bil.hdr</code>	Header file containing some meta-data relating to <code>hawk_line_mask-badpixelmethod.bil</code>
<code>hawk_line_nav_post_processed.bil</code>	Navigation synced to <code>hawk_line.bil</code> data
<code>hawk_line_nav_post_processed.bil.hdr</code>	Header file containing some meta-data relating to <code>hawk_line_nav_post_processed.bil</code>
<code>hawk_fov_fullccd_vectors.bil</code>	Hawk full CCD view vector file. Contains the Hawk CCD view vectors.
<code>hawk_fov_fullccd_vectors.bil.hdr</code>	Header file containing meta-data relating to <code>hawk_fov_fullccd_vectors.bil</code> .
<code>hawk_line.xml</code>	XML meta-data relating to <code>hawk_line.bil</code> .

B Appendix - Installation

Visit the ARSF-DAN website and download a copy of APL for your computer system from here: <http://arsf-dan.nerc.ac.uk/trac/wiki/Downloads/software>

Unzip the file contents into a directory. The windows build comes with the PROJ library included, if you are running under linux then please make sure you have PROJ installed on your system. The 3 APL files aplcorr, apltran and aplmap need to have executable rights to run them.