How to: load ENVI hyperspectral image data with MATLAB

Author: Miguel A. Veganzones

Date: 2010-03-15

ENVI files

ENVI binary files use a generalized raster data format that consists of two parts:

- Binary file ('.bsq', '.bil' or '.bip' extensions): flat binary file.
- Header file ('.hdr' extension): text (ASCII) file containing the metadata associated with the binary file, needed to load the binary data with MATLAB.

Let's see an example of an ENVI hyperspectral header file:

```
ENVI
description = {  DN to Radiance Conversion Version 2.6, (C)2001-2008 HyVista  Corp.#RADFILE=C:\Documents and Settings\hvc-fujitsu\Desktop\calibration_hymap2_17July09.txt#FACFILE=#NIR=2#REJECT=0,  32 [Mon Aug 3 15:17:57 2009]}
samples = 512
lines   = 2878
bands   = 125
header offset = 0
file type = ENVI Standard
data type = 2
interleave = bsq
sensor type = Unknown
byte order = 0
wavelength units = Unknown
wavelength = {  453.700012,  467.700012,  482.600006,  497.399994,  512.099976,  527.099976,  541.900024,
556.700012,  571.400024,  586.400024,  601.200012,  615.900024,  630.299988,  644.799988,  659.200012,
673.799988,  688.299988,  702.799988,  717.200012,  731.500000,  745.900024,  760.099976,  774.200012,
788.400024,  802.900024,  817.099976,  831.200012,  845.400024,  859.400024,  873.000000,  886.099976,
891.500000,  907.000000,  922.400024,  937.900024,  953.700012,  968.900024,  984.299988,  999.900024,
1015.099976, 1030.300049, 1045.400024, 1060.500000, 1075.099976, 1089.800049, 1104.699951, 1119.300049,
1133.800049, 1148.199951, 1162.599976, 1177.099976, 1191.300049, 1205.400024, 1219.500000, 1233.599976,
1247.699951, 1261.599976, 1275.500000, 1289.199951, 1302.800049, 1316.800049, 1330.500000, 1389.199951,
1405.000000, 1419.800049, 1434.199951, 1448.699951, 1463.000000, 1477.500000, 1491.500000, 1505.400024,
1519.199951, 1533.000000, 1546.699951, 1560.199951, 1573.599976, 1586.800049, 1599.900024, 1613.099976,
1626.300049, 1639.300049, 1652.099976, 1664.800049, 1677.500000, 1690.000000, 1702.500000, 1715.000000,
1727.300049, 1739.699951, 1751.900024, 1764.000000, 1776.000000, 1788.000000, 1799.900024, 1950.300049,
1971.500000, 1990.800049, 2010.099976, 2029.300049, 2048.500000, 2067.300049, 2085.899902, 2104.300049,
2122.699951, 2140.800049, 2158.800049, 2176.500000, 2193.500000, 2212.100098, 2229.899902, 2247.399902,
2265.199951, 2282.199951, 2299.199951, 2316.199951, 2333.199951, 2350.100098, 2366.800049, 2383.300049,
2399.500000, 2415.899902, 2432.100098, 2448.300049, 2464.199951, 2479.800049}
```

Note: The header file could contain more information not relevant for the current explanation.

The most important data here are:

- The number of lines and samples: image's spatial dimensionality
- The number of bands: image's spectral dimensionality
- The header offset: number of bytes in the binary file that we have to skip before accessing the raster data:

- The data type: the binary data codification.
  - 1: 1-byte unsigned integer
  - 2: 2-byte signed integer
  - 3: 4-byte signed integer
  - 4: 4-byte float
  - 5: 8-byte double
  - 9: 2x8-byte complex number made up from 2 doubles
  - 12: 2-byte unsigned integer
- The interleave: way in which the raster data are ordered to access them. You can find a good description of the interleave options here: http://webhelp.esri.com/arcgisdesktop/9.2/index.cfm?id=2527&pid=2519&topicname=BIL,_BIP,_and_BSQ_raster_files
  - BSQ: Band Sequential (X[col,row,band])
  - BIL: Band Interleave by Line (X[col,band,row])
  - BIP: Band Interleave by Pixel (X[band,col,row])
- The byte order: the endian-ness of the saved data:
  - 0: means little-endian byte order, format used on PC/Intel machines
  - 1: means big-endian (aka IEEE, aka "network") byte order, format used on UNIX and Macintosh machines

MATLAB 'multibandread' command

To load the ENVI's binary hyperspectral image data with MATLAB we can use the '*multibandread*' command:

```
X = multibandread(FILENAME, SIZE, PRECISION, OFFSET, INTERLEAVE, BYTEORDER)
```

Let's follow with our example above. To load the binary file 'image.bsq' we have to look at the 'image.hdr' information.

- FILENAME corresponds to the name of our binary file: 'image.bsq'.
- SIZE = [LINES, SAMPLES, BANDS] and in our example corresponds to [2878, 512, 125].
- PRECISION corresponds to the data type in the ENVI header file. In our example is 2 (2-byte signed integer). Table 1 shows some correspondences between ENVI data types and MATLAB precision values.

ENVI data type	MATLAB precision
1 – 1 byte signed integer	'int8'
2 – 2 byte signed integer	'int16'
3 – 4 byte signed integer	'int32'
4 – 4 byte float	'float'
12 – 2 byte unsigned integer	'uint16'

Tabla 1: Correspondences between ENVI data types and MATLAB precision values

- OFFSET: the same as offset parameter from ENVI header file.
- INTERLEAVE: the same as interleave parameter from ENVI header file: 'bsq', 'bil' or 'bip'.
- BYTEORDER: corresponds to byte order parameter from ENVI header file. However it has to be translated to MATLAB codification. Table 2 shows the right values.

ENVI byte order	MATLAB byteorder
0	'ieee-le'
1	'ieee-be'

Tabla 2: Correspondences between ENVI byte order and MATLAB byteorder values

So, in our example to load the binary image data we have to execute in MATLAB the following command:

```
X = multibandread('image.bsq',[2878,512,125],'int16',0,'bsq','ieee-le');
```

This will load in 'X' variable the hyperspectral cube where the first and second dimensions correspond to the spatial coordinates and the third dimension is the spectral coordinate.

Note: hyperspectral images (hyperspectral cubes) are usually huge and may run your computer out of memory. If so, please consult next section “Loading subimages”.

Loading subimages

If you are not interested in loading the full hyperspectral cube, you can load directly a subscene (spatial patch) or a determined spectral range (spectral patch) by passing some arguments to the '*multibandread*' command:

```
X = multibandread(FILENAME, SIZE, PRECISION, OFFSET, INTERLEAVE, BYTEORDER,
SUBSET, SUBSET, SUBSET)
```

SUBSET = {DIM, METHOD, INDEX} indicates the rows, columns or bands to be loaded:

- DIM = 'Row', 'Column' or 'Band'.

- METHOD = 'Direct' or 'Range'.
    - If using 'Direct', INDEX is a vector specifying the indices to read along the DIM dimension.
    - If using 'Range', INDEX is a 2 or 3 element vector [START, INCREMENT, STOP] specifying the range and step size to read along the dimension. If INDEX is 2 elements, then INCREMENT is assumed to be one.

If we want load a sub-scene of our sample image we can use:

```
X = multibandread('image.bsq',[2878,512,125],'int16',0,'bsq','ieee-le',
{'Row','Range',[1:100]}, {'Column','Range',[1:100]});
```

Then, variable 'X' will contain a 100x100x125 sub-scene corresponding to the superior-left corner of the full image.

If for example, we are only interested in the bands 10, 23 and 64 we can use:

```
X = multibandread('image.bsq',[2878,512,125],'int16',0,'bsq','ieee-le',
{'Band','Direct',[10,23,64]});
```

Then, variable 'X' will contain a 2878x512x3 matrix, where the third component corresponds to the bands 10, 23 and 64.

We can use any combination to get the information of our interest.

Visualizing hyperspectral information with MATLAB

Given that we have loaded the hyperspectral data of our interest in the variable 'X' with dimensions [M,N,P], we can do some visualizations:

- Spatial visualization: if we want to see the spatial information for a single wavelength.
- Spectral visualization: if we want to see the spectral response of a given pixel.

In the first case, spatial visualization, we have to extract from 'X' the slice of our interest. If, for example, we are intereseted in the band 34:

```
Y = squeeze(X(:,:,34))
```

and now 'Y' will contain a bidimensional matrix that we can visualize with the 'imagesc' or 'imshow' commands.

In second case, spectral visualization, the process is similar. If we are interested in the spectral response of pixel (152,256):

```
Z = squeeze(X(152,256,:))
```

and now 'Z' will contain a vector that we can visualize with the 'plot' command.