Natural Resources
Canada

Ressources naturelles
Canada

Processing EO-1 Hyperion data using ISDAS

R. Hitchcock and H.P. White

2007

Geomatics Canada

TECHNICAL NOTE 2

Canada

ISSN 1914-4229
Catalogue No. M103-1/2-2007E-PDF
ISBN 978-0-662-46408-2

A copy of this publication is also available for reference in depository libraries across Canada through access to the Depository Services Program's Web site at http://dsp-psd.pwgsc.gc.ca

A free digital download of this publication is available from GeoPub: http://geopub.nrcan.gc.ca/index_e.php

Toll-free (Canada and U.S.A.): 1-888-252-4301

Critical reviewers
L. Sun
R. Soffer

Authors

R. Hitchcock (hitchcoc@nrcan.gc.ca)
H.P. White (pwhite@nrcan.gc.ca)
Hyperspectral Working Group
Canada Centre for Remote Sensing
Earth Sciences Sector
Natural Resources Canada
588 Booth Street
Ottawa, Ontario K0A 0Y7

Publication approved by Earth Observation and GeoSolutions Division

Correction date:

CONTENTS

Processing EO-1 Hyperion data using ISDAS

R. Hitchcock and H.P. White

Hitchcock, R. and White, H.P., 2007: Processing EO-1 Hyperion data using ISDAS; Geomatics Canada, Technical Note 2, 30 p.

Abstract: With increased global interest and development of hyperspectral spaceborne and airborne sensors, high spectral and spatial Earth observation data will become more readily available. With this improvement in spectral and spatial resolution comes the need to have more rigorous image preprocessing. Spectral and spatial registration and radiometric response need to be characterized and applied more frequently, possibly on a scene-by-scene basis, depending on the stability of the sensor. This requires a system that can evaluate a data set and determine parameters efficiently and independently.

A preprocessing procedure to transform at-sensor signals to at-surface reflectance for Earth observation hyperspectral imagery has been developed at the Canada Centre for Remote Sensing, Natural Resources Canada (CCRS, NRCan). This process examines an image cube for bad pixels (stripes) and noise levels, determines spectral (smile effect) and spatial (keystone) registration per pixel, as well as evaluating the image cube for optimal signal gain and offset, and applies relevant corrections. Where applicable, a scene-based (vicarious) calibration procedure can also be applied.

This procedure has been implemented in the Imaging Spectrometer Data Analysis System (ISDAS) developed at CCRS. The following document outlines how to apply the detection and correction algorithms to derived at-surface reflectance from at-sensor radiance, using an EO-1 Hyperion image acquisition as an example data source.

Résumé : En raison d'un intérêt croissant à l'échelle mondiale et du développement accéléré de capteurs hyperspectraux satelliportés et aéroportés, les données d'observation de la Terre à haute résolution spectrale et spatiale deviendront plus facilement accessibles. Cette amélioration de la résolution spectrale et spatiale a pour corollaire un prétraitement plus rigoureux des images. Selon la stabilité du capteur, le repérage spectral et spatial ainsi que la réponse radiométrique doivent être caractérisés et plus fréquemment appliqués, peut-être même scène par scène dans certaines situations. Cela exige un système permettant d'évaluer un jeu de données et de déterminer les paramètres pertinents de manière indépendante et efficace.

Le Centre canadien de télédétection de Ressources naturelles Canada (CCT/RNCan) a mis au point, pour l'imagerie hyperspectrale d'observation de la Terre, une procédure de prétraitement permettant la transformation des signaux au capteur en réflectance à la surface du sol. Par ce processus, on examine un cube image afin d'identifier de mauvais pixels (rayures) et de définir les niveaux de bruit, on détermine le repérage spectral (effet de sourire) et spatial (déformation trapézoïdale) par pixel, tout en évaluant le cube image quant au gain et au décalage optimaux du signal, et on applique les corrections pertinentes. Lorsqu'applicable, une procédure d'étalonnage (vicariante) basée sur une scène peut aussi être utilisée.

Cette procédure a été mise en œuvre dans le Système d'analyse de données de spectromètre imageur (*Imaging Spectrometer Data Analysis System*, ISDAS), mis au point au CCT. Le présent document explique dans les grandes lignes la façon d'appliquer les algorithmes de détection et de correction afin d'obtenir la réflectance à la surface du sol à partir de la luminance au capteur, en se servant de l'acquisition d'images par le capteur Hyperion EO-1 comme exemple de source de données.

INTRODUCTION

This document is intended to provide ISDAS users with a step-by-step procedure for atmospherically correcting EO-1 Hyperion at-sensor scaled radiance in order to produce surface reflectance using the dec_13_04 version of ISDAS software. In this document, ISDAS module names appear in ***bold italic text*** e.g. ***ModuleName***. The location of where each module can be found in the ISDAS menus is shown in Appendix A. A flow chart of the EO-1 Hyperion processing procedure can be found in Appendix B.

This document is not intended to teach new users how to use ISDAS, so a basic level of familiarity with ISDAS is assumed.

For a general overview of ISDAS *see* Staenz et al. (1998), and for a more detailed description of the preprocessing steps performed in this document, *see* Khurshid et al. (2006).

For some of the modules, example parameters are shown in blue. *Note: these example parameters may not apply to every data set and care should be taken to consider other values for these parameters before executing the modules.*

In an effort to standardize output filenames for ease of file and project sharing and troubleshooting, recommended filenames are shown in **red bold font**.

ISDAS-related conventions used in this document are:

⇓ indicates a pull-down menu or pull-down menu selection e.g. ⇓ File ⇓ Close in Figure 1.

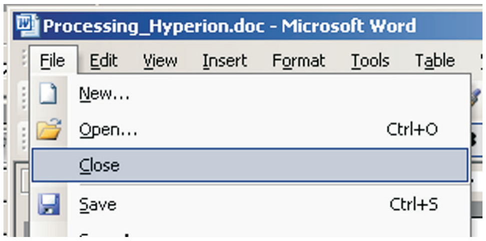


Figure 1. Example of a pull down menu with selections.

⊕ indicates a button selection (Fig. 2a, b).

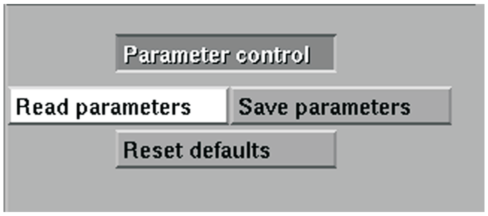


Figure 2. Example button selection options within **a)** an ISDAS module parameter window;

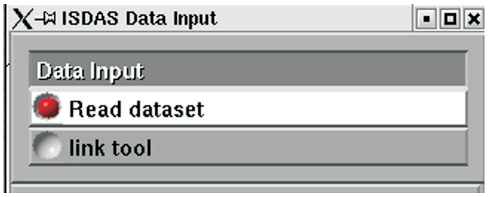
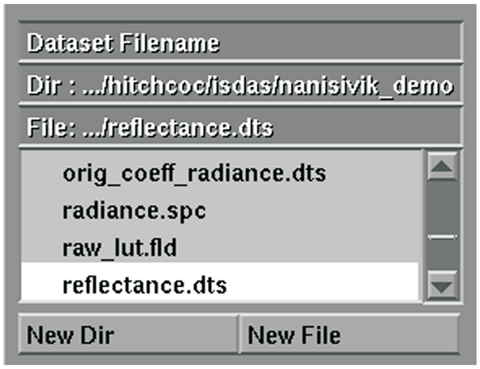


Figure 2. b) an ISDAS module dialog window.

⇒ indicates a selection by highlighting from a window list (Fig. 3).



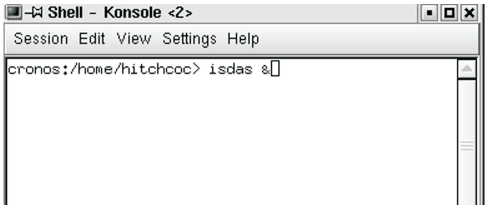
Dir : .../hitchcoc/isdas/nanisivik_demo

File: .../reflectance.dts

orig_coeff_radiance.dts
radiance.spc
raw_lut.fld
reflectance.dts

Figure 3. Example of an ISDAS module file browser and selection window.

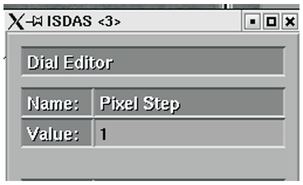
•> command indicates a Linux command from a console window (Fig. 4).



```
cronos:/home/hitchcoc> isdas &
```

Figure 4. Example of a Linux console window.

{ } indicates a type-in selection in an ISDAS module (Fig. 5).



ISDAS <3>

Dial Editor

Name: Pixel Step

Value: 1

Figure 5. Example of an ISDAS dialog type-in window.

♣ indicates a type-in window that is activated by left clicking with the cursor on the centre circle of a dial-in widget (Fig. 6).

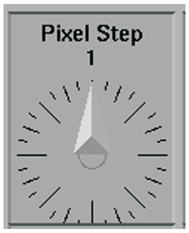


Figure 6. Example of an ISDAS dialog dial-in widget window

1. CONSIDERATIONS BEFORE STARTING

It is important to note that the EO-1 Hyperion sensor is physically made up of two detector arrays. Detector #1 spans bands 1–70 (355–1057 nm), referred to here as the VNIR (visible-near infrared) while detector #2 spans bands 71–242 (851–2577 nm) and is referred to as the SWIR (short-wave infrared). There is spectral overlap between the detectors; however, the last 13 overlap bands of the VNIR are zeroed (no data); so effectively there are only two VNIR bands (56–57), and two SWIR bands (77–78), which overlap. Typically bands 56 and 57 are discarded and the first two good bands of the SWIR are used. This is due to considerations for wavelength shift and smile detection.

The following questions should be answered before starting:

- Is there ground spectrometer data (GER, ASD, etc.) useful for scene-based (vicarious) radiometric calibration?
- Is there a particular ROI (region of interest) to focus on or will the entire scene be processed?

Note: certain modules (e.g. noise removal) may have trouble handling more than 2000 lines so the flight line may have to be split and processed in two parts.

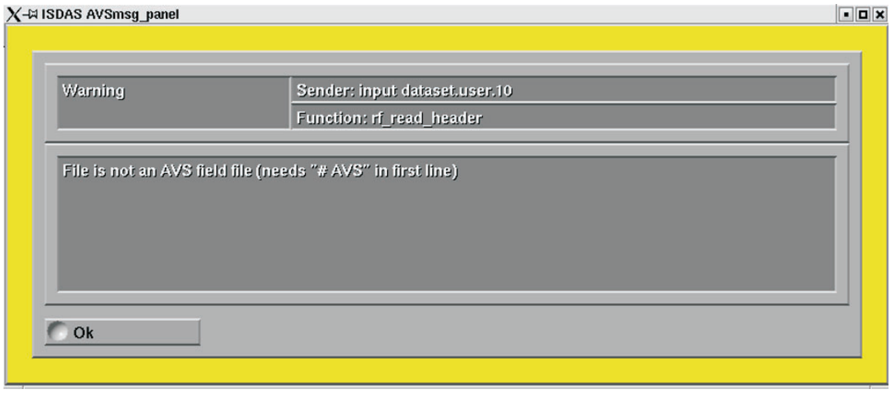
- Is the scene primarily over water? Significant bodies of water may detrimentally effect the water-vapour estimation and possibly the smile detection; therefore spatial cropping may be necessary.

A typical EO-1 Hyperion scene of 3000 to 3500 lines will require approximately 5.5 Gigabytes of space if the output from all of the intermediate steps is kept (recommended).

2. CONVERTING THE DATA FROM AT-SENSOR SCALED RADIANCE TO SURFACE REFLECTANCE

2.1. Initial data verification

- Create an appropriately named working directory. Avoid nesting directories too deeply and using long directory names or filenames. *If the combination pathname and/or filename is too long, this will result in the error message* (Fig. 7) *while reading in the data in ISDAS.*
- Verify that the scene ID on the surface of the CD-ROM (e.g. EO1H##…##) is the same as that on the CD-ROM cover.
- Load the CD-ROM and do a directory listing. For EO-1 Hyperion data processed by the USGS there will be a file named [SCENE_ID] ***.sum** as well as a directory with the same name as the scene ID. Data processed prior to 2005 will not have the ***.sum** file.



ISDAS AVSmsg_panel

Warning

Sender: input dataset.user.10

Function: rf_read_header

File is not an AVS field file (needs "# AVS" in first line)

Figure 7. Example of an ISDAS error display window.

- Copy both to the working directory.
- The ***.sum** file is of little use so go to the scene ID subdirectory. There will be three files: [SCENE_ID]**.hdr**; [SCENE_ID]**.L1R**; and [SCENE_ID]**.met**. The ***.L1R** file is the at-sensor radiance cube, typically about 600 Megabytes in size.
- Open the header (***.hdr**) file in a text editor. Make note of the X (samples), Y (lines), and Z (bands) dimensions. Typically EO-1 Hyperion has 256 samples (across track pixels) and 242 bands. Verify the data type is 2 (16 bit signed) and the interleaving is band interleaved by line (BIL). The nominal band centres and bandwidths for all 242 bands, measured in nanometres, are found at the end of the file. If access to a previously created EO-1 Hyperion sensor file (***.ssr**) is not an option, then this information can be used to create one (step 2.5). The bandwidths and band centres contained here are nominal values and the smile detection and smile correction step will result in more accurate values.
- Open the metadata (***.met**) file in a text editor. The first line is the scene ID, which should be the same as the scene ID on the CD-ROM. The first three characters will always be E01. The next three characters are the Landsat7 path containing the scene. The next three characters are the Landsat7 row containing the scene. The next four characters are the year. The next three characters are the day-of-year date i.e. Jan 1st = 001. Convert this to a calendar date and verify that the date corresponds with what was ordered. The next two lines in the file contain the latitude and longitude of what was ordered (*not necessarily what is on the CD-ROM*). Verify that the scene centre is correct and that it lies within the four reported corner latitudes and longitudes. Ideally it should be the exact scene centre, but it may not be. A visual inspection at step 2.3 will verify that the desired ROI is actually contained in the scene.

Next, make note of the "HYP File Start Stop Duration" times. This is the scene acquisition time in GMT (Greenwich mean time). The duration should be 15.000 s. Calculate the scene centre time as half way between the start and stop times.

It is highly recommended that all of this information, as well as all of the settings used at each processing step, and the ISDAS version number be kept in a digital file. Each scene should have its own file which is stored in the working directory for that scene.

2.2. Convert the raw data from HDF format to ISDAS-compatible format

Hyperion imagery is provided in a hierarchical data format which cannot be ingested directly into ISDAS, but which can be easily ingested and manipulated by both PCI and ENVI into a format that ISDAS can handle. The HDF format data contains five layers of data, and each layer has to be stripped off and saved as individual files. The procedures using both PCI and ENVI are included.

2.2.1. Procedure using ENVI

- Start ENVI.
- Load the data into ENVI.

  ⇓ File ⇓ Open Image File *type in directory containing raw data*

  ⇒ select the *** L1R** file

  ⊕ OK ⊕ Select All Items ⊕ OK ⊕ BIL ⊕ OK

  at this point ENVI will prompt the user as to the interleaving for the Flag Mask data

  ⊕ BIL ⊕ OK

Click on the minus sign to the left of Data Set #5, to minimize the details. For ***.L1R** files created after September 2002, there will be five data sets as part of the hierarchical data structure (Fig. 8):

#5 **Flag Mask:** indicate problem pixels;

#4 **Gain Coefficients:** (*Note: for* ***.L1R** *data, these gains have already been applied*). This data will be used for noise removal;

#3 **Bandwidths**: not the average values in the ***.hdr** file, but detailed information for smile correction;

#2 **Centre Wavelengths**: not average values in the ***.hdr** file, but detailed information for smile correction; and

#1 **Hyperspectral Data Cube**: (in scaled radiance values).

- Expand Data Set #1 (click on the 'plus' sign if necessary) and verify that there are 242 bands. The bottom slider may have to be moved to the right to see the band numbers.
- Visually inspect the image to make sure that there is data and that the scene contains the ROI.

  ⇒ highlight band 40 of Data Set #1 ⊕ Grey Scale
  ⊕ Load Band

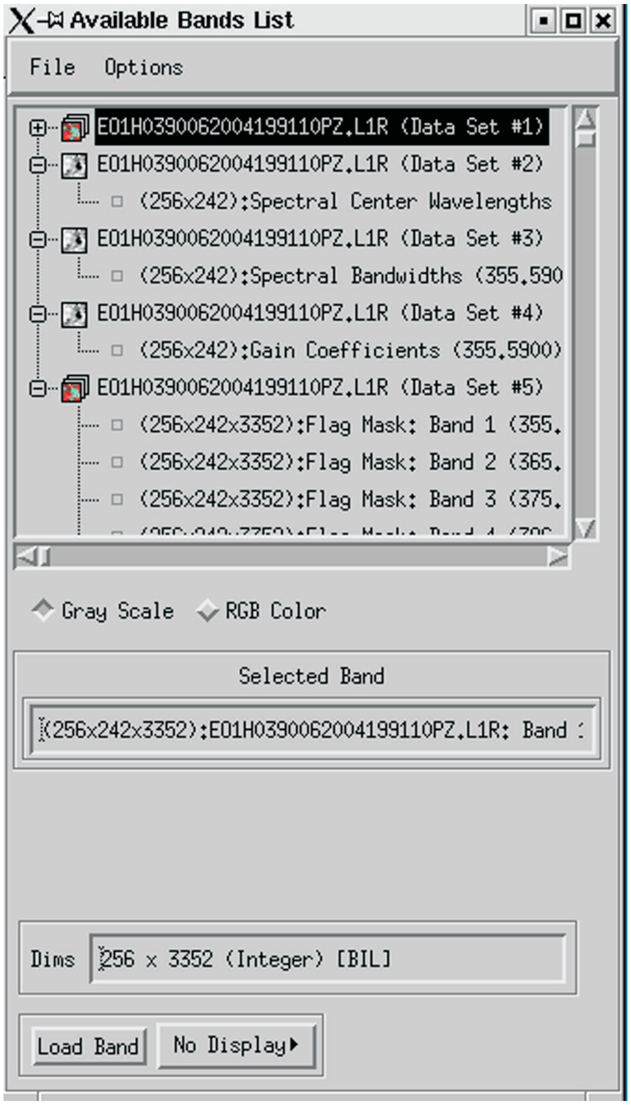
Use the cursor and the red Scroll box (Fig. 9) to scroll through the entire flight line. Linear enhancement may be needed to aid in inspection

- Save the radiance data cube (Data Set #1) as BIL, unsigned integer.

  From the main ENVI menu bar,

  ⇓ File ⇓ Save File As ⇓ ArcView Raster (bil) ⇒ Data Set #1 ⊕ OK

  working directory/**radiance.bil**
  Output Data Type ⇓ Unsigned Integer
  ⊕ OK


Available Bands List

File Options

- E01H0390062004199110PZ.L1R (Data Set #1)
- E01H0390062004199110PZ.L1R (Data Set #2)
  - (256x242):Spectral Center Wavelengths
- E01H0390062004199110PZ.L1R (Data Set #3)
  - (256x242):Spectral Bandwidths (355.590
- E01H0390062004199110PZ.L1R (Data Set #4)
  - (256x242):Gain Coefficients (355.5900)
- E01H0390062004199110PZ.L1R (Data Set #5)
  - (256x242x3352):Flag Mask: Band 1 (355.
  - (256x242x3352):Flag Mask: Band 2 (365.
  - (256x242x3352):Flag Mask: Band 3 (375.

○ Gray Scale ○ RGB Color

Selected Band

(256x242x3352):E01H0390062004199110PZ.L1R: Band :

Dims 256 x 3352 (Integer) [BIL]

Load Band | No Display ▸

Figure 8. An ENVI data set selection window.

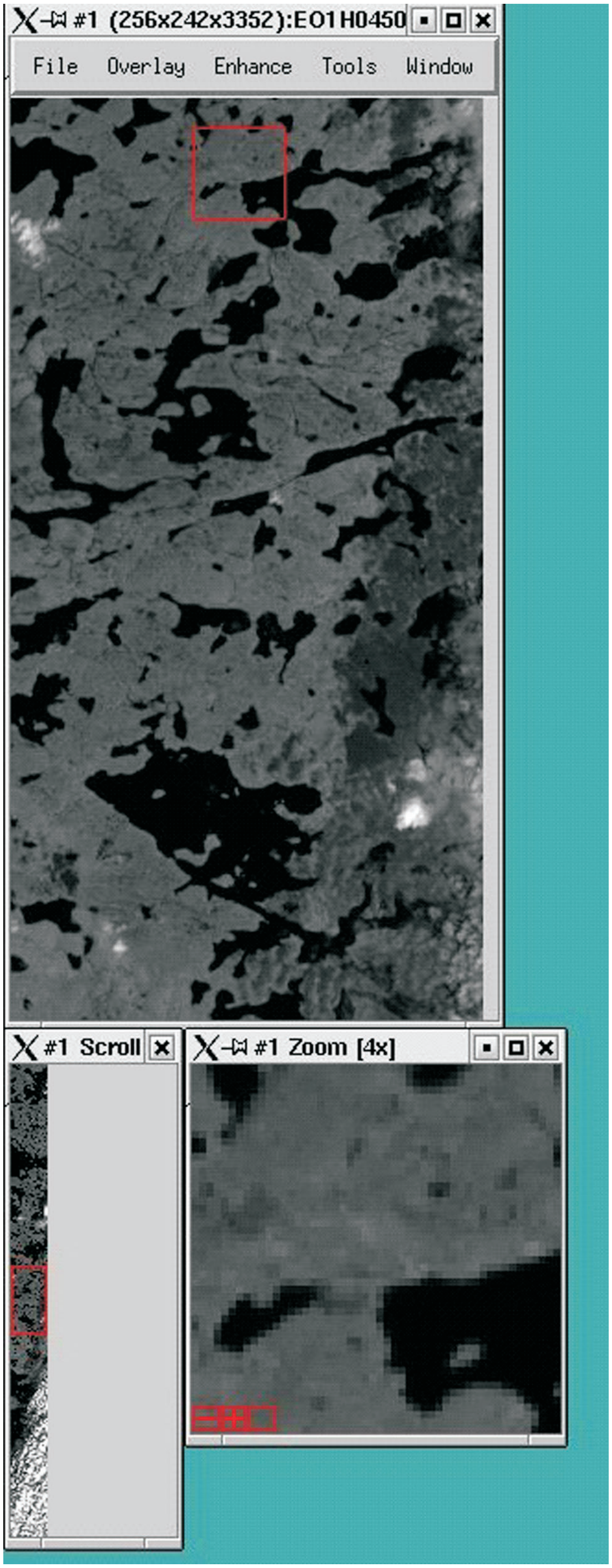


Figure 9. ENVI data set display windows.

Do a directory listing of the working directory to make sure that the following three files were written: **radiance.fld**, **radiance.stx**, **radiance.hdr**.

- Save the Spectral Centres Wavelengths (Data Set #2) as ASCII text files.

  ⇓ File ⇓ Save File As ⇓ ASCII ⇒ Data Set #2

  ⊕ OK working directory/**bandcentres.txt**

  ⊕ OK

*Note: Band centres and bandwidths are provided for EO-1 Hyperion imagery on a per pixel basis to provide information on the sensor smile. The mean values (per band) for those not interested in applying this smile correction can be found in the *.hdr file. Plotting the band centres per pixel for a specific sensor channel will demonstrate the smile effect as reported with the data.*

- Repeat the above procedure to save Data Set #3 and Data Set #4 as ASCII files, named **bandwidths.txt** and **gains.txt**, respectively.
- Data Set #5, the Flag Masks which mark the 'bad' pixels is not needed. This information is not needed for processing because the ***Auto Destriper*** tool will automatically fix all black stripes in step 2.11.
- Now that all the files are saved, ENVI may be closed.

ENVI>*exit*

2.2.2. Procedure using PCI Geomatica (version 9.1)

- Run Geomatica and open Focus
- In Focus, open the ***.L1R** file

⇓ File ⇓ Open navigate to desired directory

⇒select the ***.L1R** file

⊕ Open

There are five files, the hyperspectral image cube and four others. All will be opened eventually.

⇒Select the image cube (labelled ***.L1R**)

⊕ OK

To view an approximate RGB image of the file, click on the 'Maps' tab in Focus. Open all levels of the tree to view the three default image layers being used. Right click on the image layer and change the layer being viewed.

Red: 30; Green: 20; Blue: 10

The 'Enhancement' may need to be reset. Try 'Adaptive' and go from there.

Click on the 'Files' tab

Right click on the filename (top of the directory tree) and click on 'Translate (Export)**'**.

Destination file: **radiance.bil**

Output format: BIL:Arc/Info BIL

Source layers: Select All > Add > Export

- The fifth data set, **FlagMask.bil**, need not be saved.
- The other three data sets (**BandCentres.txt**, **BandWidths.txt**, and **Gains.txt**) can be saved as ASCII files. To do this, use the following procedure.

⇓ File ⇓ Open navigate to desired directory

⇒select the ***.L1R** file

⊕ Open

Again, there are five files, the hyperspectral image cube and four others. Open the file of interest.

Click on the 'Files' tab

Right click on the filename (top of the directory tree) and click on 'Algorithm Librarian'.

Select the NUMWRIT module.

Select the only active channel from the input layer(s)

Input params 1

Output **form** Data256

Text Output File (one of **bandcentres.txt**, **bandwidths.txt**, and **gains.txt** as appropriate).

Click on Run Repeat for the other two files.

Now that all the files are saved, PCI Geomatica may be closed.

2.3. Import the data cube into ISDAS and save in ISDAS format

- Start ISDAS

> *isdas &* ⊕ Network Editor

- Use the ***Import Multispectral*** module to load the radiance cube.

⊕ Read multispectral data cube ⇒ **radiance.bil**

⊕ short ⊕ band interleaved by line

♣ width 256 ♣ # of bands 242 ♣ # of lines (*get this value from the viewing the* ***.hdr** *file in a text editor*) ⊕ Read File

- Connect the ***2D Tool*** as shown in Figure 10.

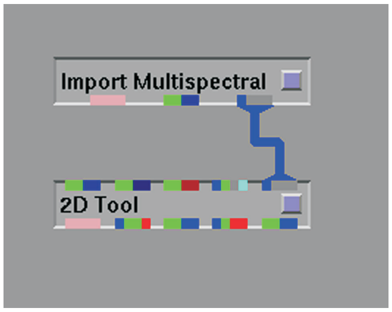


Figure 10. Network to import non-native data cubes to ISDAS and save in native ISDAS format.

Using the ***2D Tool*** controls, visually inspect a few bands to make sure the scene contains the region of interest (ROI).

*Note: the latitude and longitude values in the *.met file is the user requested latitude and longitude, so it is wise to verify that the scene actually covers the ROI. To do this, it helps* *to either have prior knowledge of the ROI or to have a map, aerial photograph, GIS, or digital map (such as the Radarsat Swath Planner) of the ROI.*

To aid inspection, move the cursor to the right hand edge of the ***2D Tool*** image window, [Shift]/hold-down-centre-mouse-and-move-right-to-left in order to make the entire flight line visible in the ***2D Tool*** window.

To select the bands to view:

⊕ select slice   Band number 8
[enter] *Bands 1 though 7 contain no data.*

- The ***Tiler Tool*** will be used in step 2.7 to view thumbnails of all of the bands simultaneously in order to quickly assess data quality. Trying to display the entire flight line as a thumbnail will result in too much decimation so it is necessary to use just a square 256 pixel x 256 line ROI that is ideally cloud-free and over land. Check to see if the first 256 lines are cloud free and over land; if they are not, use the cursor to identify a 256 pixel x 256 line ROI that is.

⊕ Roi, cursor and transect

⊕ Roi and Cursor operation

⊕ Roi and Cursor operations

⊕ pick crosshair

Click on the scene in the ***2D Tool*** window and note the (x,y) co-ordinates that are displayed in the ***2D Tool*** control panel.

- Check for spatial shift in the SWIR. View one of the SWIR bands (e.g. band 150) to see if columns after pixel 128 have been shifted down by one pixel in the line direction (Y direction). To best see this, zoom in and look at the very first line of the cube. If the shift does exist, it will look like Figure 11, in this case the ***spatial shift*** module will be used to fix the shift in step 2.10.

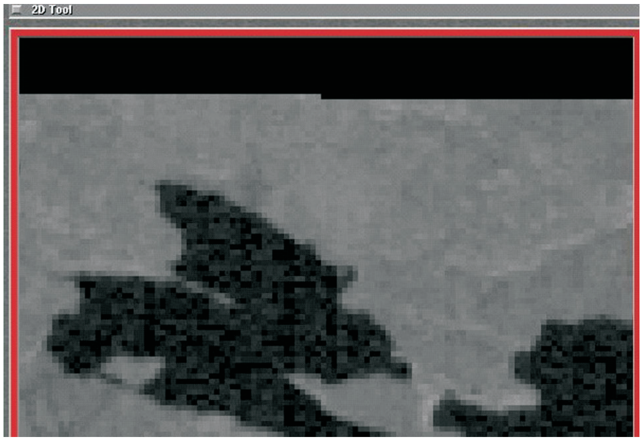


Figure 11. Example of the EO-1 Hyperion subimage vertical shift at pixel 128.

- Check for spatial misalignment between the VNIR and SWIR detectors. Connect the ***Flicker_Tool*** as shown in Figure 12.

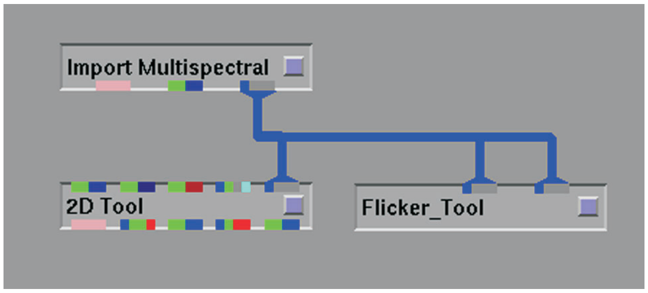


Figure 12. Network used to implement the Flicker_Tool module.

⊕ Flicker Tool

Right cube band number 50

Left cube band number 90

Flicker delay 5

⊕ Activate

Look at linear or point objects at the top and bottom of the image and in the first 128 columns only, to determine if there is a rotational shift between the VNIR and SWIR. It will be necessary to zoom in to see the shift. This can be done while ***Flicker_Tool*** is running, as can the bands to view and the delay time. Historically with EO-1 Hyperion, the SWIR bands were rotated clockwise (down to the right) with respect to the VNIR. Verify that this is the case.

If any spatial offset or rotational misalignment exists, this will be fixed at step 2.12.

- In order to get a feel for the type of land cover in the scene, create an RGB using the ***2D Tool***. Choose RGB bands (30, 20, 10) corresponding to wavelengths (650 nm, 550 nm, 450 nm) to get a true colour approximation. The clamp_min and clamp_max for each colour may have to be adjusted to perform a contrast stretch. The false colour RGB combination of bands (208, 52, 28) that represent wavelengths (2234 nm, 874 nm, 630 nm) is also quite useful for distinguishing between major land cover types.

- To save a copy of the RGB image in JPEG format, for example, use the import command in Linux, or another screen-grab utility.

>*import* **rgb.jpg**

The cursor will turn to a crosshair. Place it at the top left of the desired window, hold down the left mouse button, drag cursor to define the window and let go of the mouse button to save the file. To send it to the colour printer, it may be necessary to save it as a postscript **rgb.ps** file as well.

- Save the data cube in native ISDAS format using the ***Import Multispectral*** module.

  ⊕ Save output field

  ⇒ select working directory

  ⊕ New File

  radiance.fld

  ⊕ OK

Verify that the file has been written out to the correct directory and the file size is appropriate. It should be slightly larger in size than the **radiance.bil** file.

2.4. Create the mission (*.msn) file

- Use a text editor to manually create the mission file as per the example shown below.
- Save the file as **[SCENE_ID].msn**.
- In step 2.1 the ***.met** file that came with the raw data was viewed and the date, GMT time and scene centre latitude and longitude was noted. This information has to be entered into this mission file. If the ROI within the scene is small and its latitude and longitude are known, then this position can be entered; however, for the purpose of atmospheric correction, knowing the latitude and longitude of any point in the scene is sufficient.
- The altitude is always 705 km for EO-1 Hyperion.
- Calculate the platform heading using the following formula:

$$heading = 180° - \sin^{-1}[\cos(orbital\_inclination)(/\cos(latitude)]$$

where the orbital inclination = 98°

- Bring up a text editor and make the appropriate changes to the mission file. Note that the date is in the format YYYY-MM-DD. The tilt angle refers to the forward (or backward) pitch of the sensor along the flight line, and not the roll of the sensor across the flight track and so for EO-1 Hyperion it will always be 0.0.
- Example ***.msn** file:

  Date = 2002-04-22

  GMT = 15:50:00

  Latitude = 10.841010

  Longitude = -85.619830

  Heading = 188.146350

  Altitude = 705.0

  TiltAngle = 0.0

2.5. Create the sensor (*.ssr) file

- If a sensor file template is unavailable to edit, a new one will have to be created using the ***Hyperion sensor*** module which strips the information from the **[SCENE_ID].hdr** file.
- Create the network shown in Figure 13.

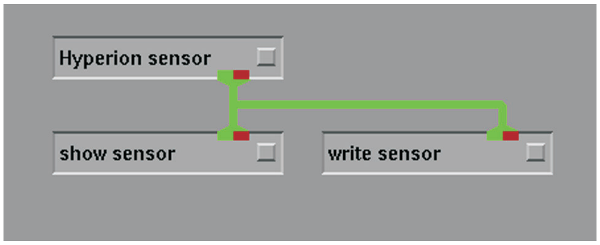


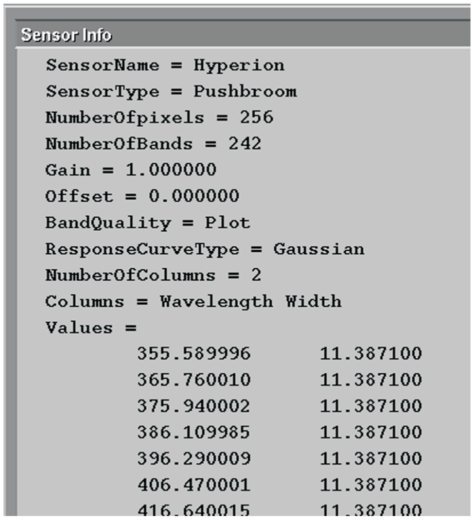
Figure 13. Network used to create an EO-1 Hyperion sensor file.

- Use ***Hyperion sensor*** to read in the **[SCENE_ID].hdr** file.

  ⊕ Hyperion sensor ⇒ **[SCENE_ID].hdr**

*Note: if either the ***.met** file or ***.L1R** file is selected, an error message will come up saying 'wrong file type'.*

- Use ***show sensor*** to view the resultant ***.ssr** file (Fig. 14). It should look like 'Error! Reference source not found'.



Sensor Info

```
SensorName = Hyperion
SensorType = Pushbroom
NumberOfpixels = 256
NumberOfBands = 242
Gain = 1.000000
Offset = 0.000000
BandQuality = Plot
ResponseCurveType = Gaussian
NumberOfColumns = 2
Columns = Wavelength Width
Values =
            355.589996       11.387100
            365.760010       11.387100
            375.940002       11.387100
            386.109985       11.387100
            396.290009       11.387100
            406.470001       11.387100
            416.640015       11.387100
```

Figure 14. Example of a sensor file for an EO-1 Hyperion data set.

- Use ***write sensor*** to write out the ***.ssr** file as **hyperion.ssr**.
- The resultant ***.ssr** file will have to be edited with a text editor as outlined below.

- Remove the Gain and Offset lines.
- All of the band-quality flags will be automatically set to PLOT at this point, but at step 2.9 they will be modified appropriately.
- The initial and final swath angles will need to be edited. To determine these angles, first obtain the instrument look angle (*not the sensor roll angle*) for the specific data cube. If this information is not included in the ***.met** file, contact the United States Geological Survey. Once the instrument look angle is obtained, add and subtract 0.312° (½ the swath width of 0.624°) to get the initial and final swath angles, respectively.
- *Note: for a southbound platform such as EO-1 Hyperion, a positive look angle means the instrument is pointing to the west.*
- Edit the sensor file with this information.
- The sensor file contains the nominal wavelengths and bandwidths for all 242 bands. View the ***.hdr** file that came with the raw data and verify that the wavelengths and bandwidths in it are very similar to those in the sensor file. As mentioned, these values are nominal (average) values and more accurate values will be recalculated with the smile detection (step 2.20).
- Example *.**ssr** file: SensorName = Hyperion

  SensorType = Pushbroom

  NumberOfpixels = 256

  InitialSwathAngle = 1.6452

  FinalSwathAngle = 1.0212

  NumberOfBands = 242

  ResponseCurveType = Gaussian

  NumberOfColumns = 3

  Columns = Wavelength Width Quality

  Values =

  355.59 11.3871 MASK

  ↓

2.6. Create the dataset (*.dts) file

- Using a text editor, manually create a ***.dts** file as per the example shown below and save it as **radiance.dts**.
- Edit the file with the appropriate pointers to the radiance data cube, the sensor file, and the mission file.
- The gains are scale factors necessary to convert the scaled radiance values in **radiance.fld** to at-sensor radiance with units of µW/cm^2/sr/nm. There is a gain value for each band, resulting in 242 gain lines. The VNIR detector bands (1–70) have a gain of 1/400 = 0.0025, whereas the SWIR detector bands (71–242) have a gain of 1/800 = 0.00125.
- Example ***.dts** file: SpectralCube = radiance.fld

  SpectralCubeUnit = Radiance

  NumberOfBands = 242

  Offset = 0.0

  Sensor = hyperion**.ssr**

  Mission = [SCENE_ID].msn

  NumberOfColumns = 1

  Columns = Gain

  Values =

  0.00250

  ↓

  0.00125

  ↓

2.7. Inspect each band

- Create the network shown in Figure 15.

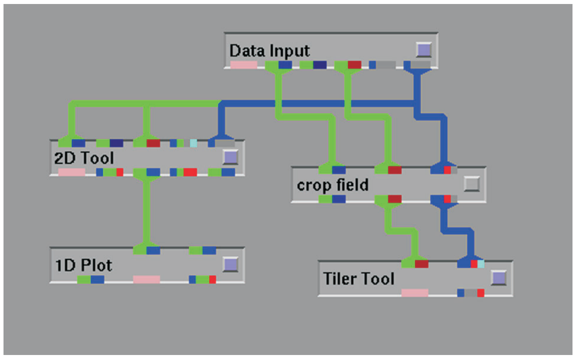


Figure 15. Network used to examine image bands.

- Use ***Data Input*** to read in **radiance.dts**.
- Using the ***2D Tool***, view band 50 to make sure that the data has been loaded properly.
- Use ***crop field*** to crop the cube in the Y direction to leave only a 256 pixel x 256 pixel cube. In step 2.3 an ROI with 256 lines of cloud-free data was identified. Ideally this would be the first 256 lines of the scene.

  To use lines 1001 to 1256 for example, do the following:

  ⊕ Y (remove line)

  from line 1

to line 1000

⊕ add to list

from line 1257

to line 3000

⊕ add to list

⊕ crop cube now

- Use the ***Tiler Tool*** to view all 242 bands simultaneously as thumbnails.

  ⊕ tiler options

  from band 0

  to band 241

  ⊕ view bands now

  The thumbnails will be displayed as in Figure 16.

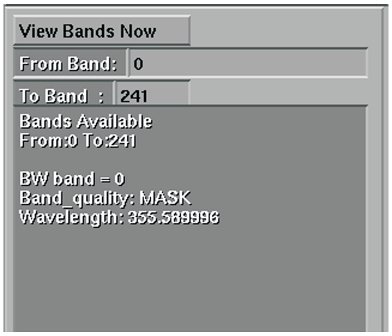
- Position the windows so that the thumbnails and also the zoom-in window called

  ‘BW/RGB Image’, can be seen.

- Verify that bad bands (all zeroes or very noisy) have been MASKed or SKIPped.

  ⊕ pick crosshair

Normally the first seven bands are all zeroes (black). Left click on the first zero band thumbnail. It should be highlighted in white and the information should be listed in the control panel (Fig. 17). *Note: the bands start counting from zero ’0’, not one ‘1’.* In ISDAS, the convention is ‘bands’ start counting from one and ‘slices’ or slice planes start counting from zero. For the remainder of this document, ‘band’ assumes that the first band is 1.



View Bands Now

From Band: 0

To Band : 241

Bands Available
From:0 To:241

BW band = 0
Band_quality: MASK
Wavelength: 355.589996

Figure 17. Tiler Tool module control panel.

Verify that the band quality flag in the ***.ssr** file is MASK. Repeat for all of the first zero bands and make note of any that are not MASKed.

Select the first visibly good band and verify that its band quality is PLOT. When the thumbnail is selected, the band will also appear in the zoom-in window in greater detail so data quality can be assessed for noise and stripes.

Normally the last two good bands of the VNIR are bands 56 and 57. Bands 58 through 76 are usually zeroes and they should have band quality SKIP.

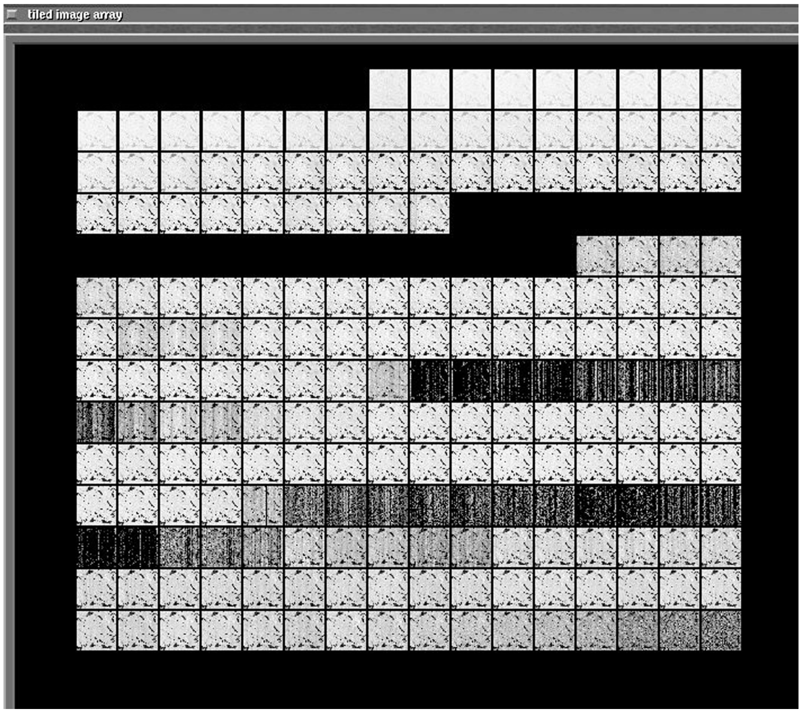


Figure 16. Thumbnail display produced by the Tiler Tool module.

Check the data quality in the 1400 nm water vapour absorption region. Typically bands 120 to 133 are noticeably poor quality (noise and striping) and should be MASKed, but band 119 is on the edge. By default it is MASKed out in the **hyperion.ssr** template, but it can be changed at step 2.9 if desired.

Repeat this for the 1900 nm water vapour absorption region. Typically bands 165 through 182 are poor quality and should be MASKed.

Bands 221 through 224 are typically bad and should be MASKed. Bands 225 through 242 are normally zeroes and should be MASKed.

The first six bands of the second detector array, bands 77–82, are typically noisy; however they are needed in the wavelength shift and smile detection. As well, bands 98–100 in the 1130 nm water vapour absorption region and bands 183–186 in the CO_2 region are typically noisy. The destriping and noise removal steps of preprocessing will increase the quality of these last three regions.

2.8. View radiance spectra

- From the previous step, the ***1D Plot*** should already be connected to the ***2D Tool***.
- Using the ***2D Tool***, select a nonzero bands with little noise (e.g. Band 40).
- Use the cursor to select pixels in the ***2D Tool*** and view their spectra in the ***1D Plot***.

⊕ Roi, cursor and transect

⊕ Roi and Cursor operation

⊕ Roi and Cursor operations

⊕ pick crosshair

⊕ update spectra

Look for any spectral anomalies such as a drastic change in magnitude between the VNIR and SWIR detectors at 915 nm. It may also be helpful to look for wavelength shifts for the 760 nm (O_2), 820 nm, 940 nm, and 1130 nm water vapour absorption features to see if there is a problem with a large wavelength shift. Select spectra from different land cover types. Verify that the average and maximum radiance values make sense for the spectra; expect radiance values of between 0 and 10 for vegetation. If the values are much higher or lower than this, check the gains in the ***.dts** file or the data may just be bad due to detector problems. A typical vegetation spectra should look something like Figure 18.

2.9. Edit band quality

- In step 2.7, all of the bands were viewed and certain bands were identified whose band quality flag in the ***.ssr** file has to be changed.

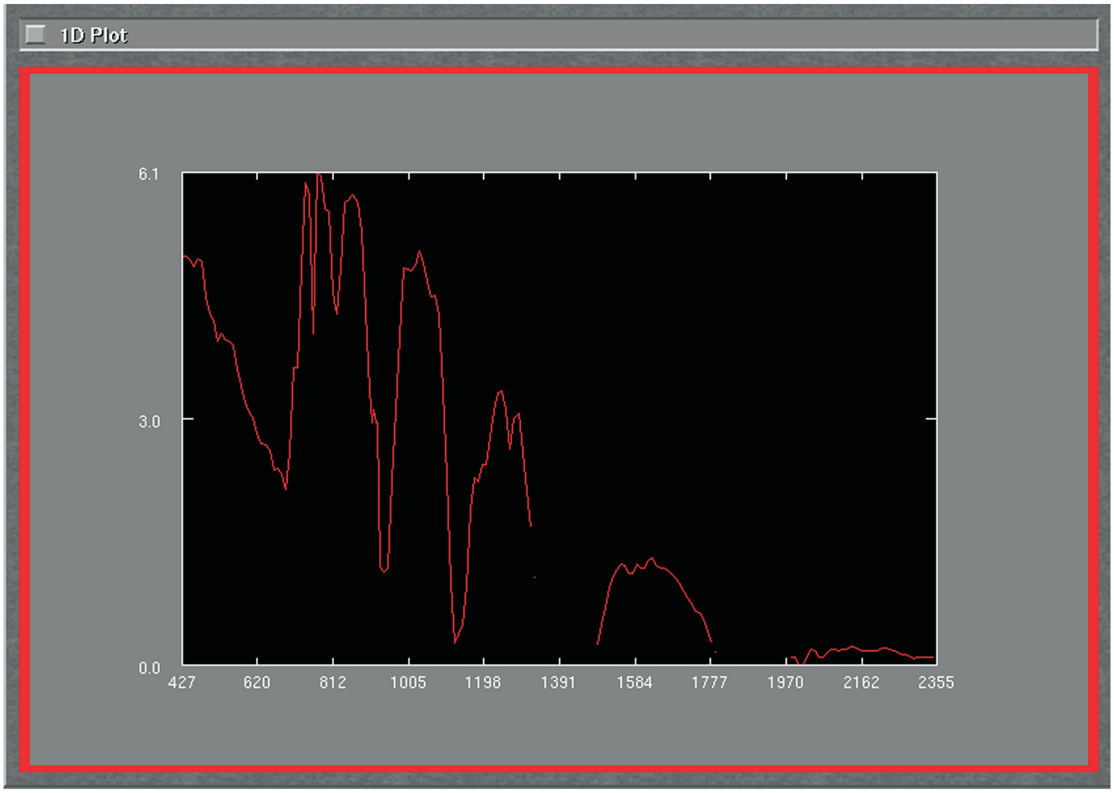


Figure 18. A typical vegetation radiance spectrum.

- If only a few bands need to be modified, then a text editor can be used to quickly edit the ***.ssr** file, otherwise, use the network and steps below.
- There are three band quality flags: PLOT, MASK, and SKIP. PLOT means that ISDAS modules will process this band. MASK means that the modules will not process this band and it also will not be plotted in the ***1D Tool***, resulting in a break in the spectrum. SKIP means that the band will not be processed and it will not be plotted, however, instead of a break in the spectra, the ***1D Tool*** will join the previous and next PLOT bands with a straight line.
- Connect the ***edit_band_quality*** and ***write_sensor*** modules to the existing network as shown in Figure 19 or if the network has been cleared, create the network in Figure 20.
- Edit the band quality.

  ⊕ edit band quality

  ⊕ select either single value or range of values

  ⊕ select either wavelengths or bands

  type in the wavelength(s) or band(s)

  ⊕ Modify band quality

  Repeat above step for each range of bands to modify.

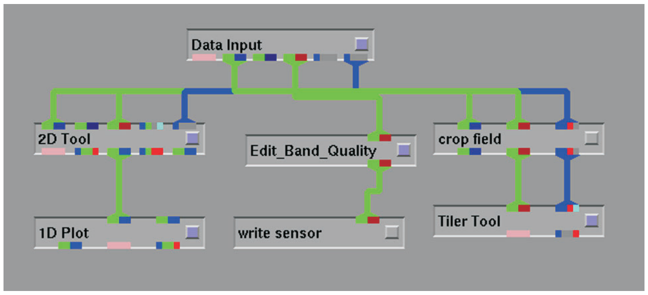


Figure 19. Expanded network to edit band-quality flags in the sensor file.

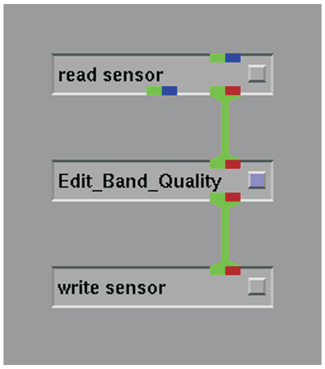


Figure 20. Stand-alone network to edit band-quality flags in the sensor file.

When done,

⊕ output modified sensor

⊕ write sensor ⇒ **hyperion.ssr**

⊕ overwrite

Figure 21 shows an example of how to use the ***edit_band_quality*** module.

Edit the ***.ssr** file in a text editor and remove the Gain and Offset lines.

2.10 Fix the spatial shift in the SWIR

- If, in step 2.3, the SWIR bands had a spatial shift of one pixel in the Y direction at column 128, then the ***spatial shift*** module can be used to fix this. This module simply shifts the right-hand side of the SWIR bands up by one pixel.
- Connect the network as shown in Figure 22.
- Use ***Data Input*** to read in radiance.dts.
- Run the ***spatial shift*** module.

  ⊕ spatial shift

  ⊕ Hyperion X-Track Fix

  First SWIR band 71 Section point 128

  ⊕ Fix Shift Now
- View some SWIR bands of output cube in the ***2D Tool*** to verify that the spatial shift worked correctly. Zoom in close to make sure the offset is fixed. Using ***Data Output***, write out the three files as **spatialshift.fld**, **spatialshift.ssr**, and **spatialshift.dts**.

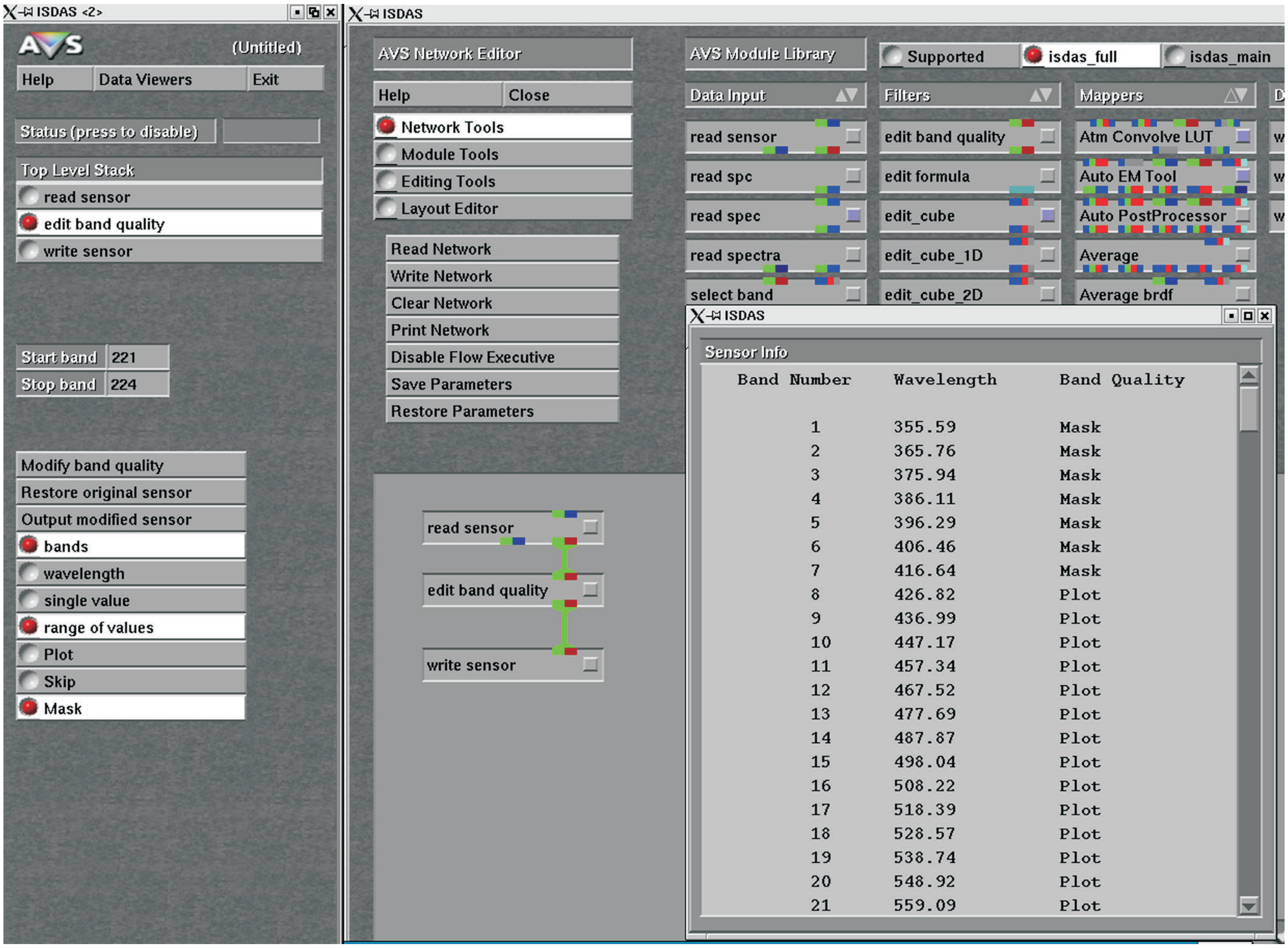
Note: whenever files are saved or written, always verify that all of the files have been written to the desired directory and that the file sizes make sense.

- Edit the ***.dts** file to remov absolute paths (directoriet) gor mission, sensor, and data cube, keeping only the filenames.
- Edit the ***.ssr** file to remove the Gain and Offset lines.
- \>clean

Note: as commented on at the beginning of this document, it is highly recommended that a digital file be created that keeps track of the information and settings used at each processing step. This will greatly help when revisiting and reviewing the data.

2.11. Automatic destriping

EO-1 Hyperion data typically has many bands with vertical striping problems. Historically there has not been a problem with horizontal striping with this sensor. The ***Auto***



Start band 221

Stop band 224

Band Number	Wavelength	Band Quality
1	355.59	Mask
2	365.76	Mask
3	375.94	Mask
4	386.11	Mask
5	396.29	Mask
6	406.46	Mask
7	416.64	Mask
8	426.82	Plot
9	436.99	Plot
10	447.17	Plot
11	457.34	Plot
12	467.52	Plot
13	477.69	Plot
14	487.87	Plot
15	498.04	Plot
16	508.22	Plot
17	518.39	Plot
18	528.57	Plot
19	538.74	Plot
20	548.92	Plot
21	559.09	Plot

Figure 21. Example display while implementing edits to the sensor file.

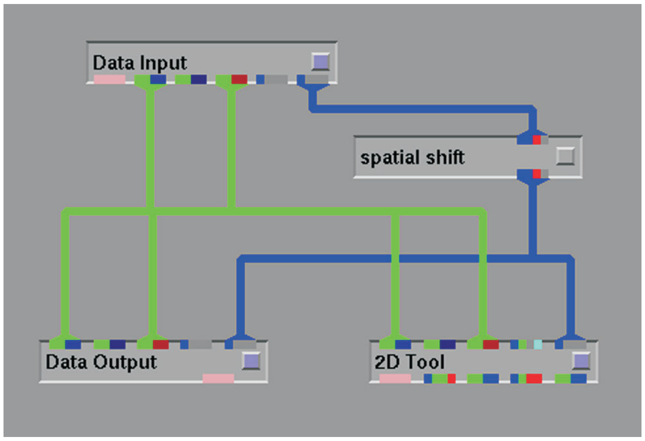


Figure 22. Network to correct the EO-1 Hyperion SWIR detector subimage shift.

Destriper module automatically identifies and fixes vertical stripes and black columns (dead pixels) based on natural variations of the adjacent pixels using spectral moment matching (Sun et al., in press). The old destriping module, ***Unstriping***, still exists and allows one to manually destripe using spatial and/or spectral techniques; however, this level of preprocessing is not usually needed.

- Connect the network shown in Figure 23.
- Use ***Data Input*** to read in **spatialshift.dts.** The ***Auto Destriper*** will automatically start collecting statistics. This may take a few minutes.
- Arrange (move and shrink) the four SMGraph windows and the 'before' and 'after' destriping image windows so that all of them can be seen simultaneously.
- Using the plane slicer select a plane to view that has lots of vertical stripes e.g. plane 76 (band 77).

♣ chosen plane 76.

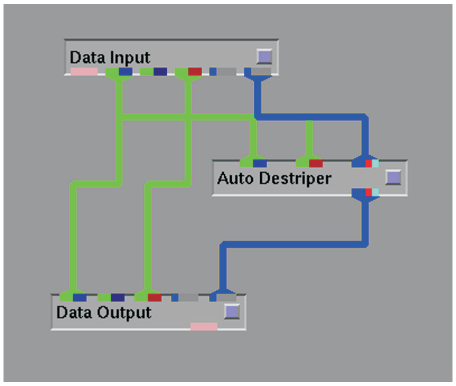


Figure 23. Network to evaluate and remove the effects of stripes (lines or columns of data with unique gain and offsets) within a data set.

After selecting the plane, the image should appear in one of the image windows. In this image the striping should be apparent. The other window should be blank. This is where the destriped image will be displayed. Four graphs should appear in the SMGraph windows. In EO-1 Hyperion data, only the along-track columns (vertical stripes) have problems so SMGraph windows 2 and 3 can be minimized or shut down.

Note: the contents in an SMGraph window will be wiped clean when a window in moved in front of it. If this happens, refresh by choosing a different slice plane to display and then reselecting the desired plane.

SMGraph <1> shows the column means. The stripes will show up in this window. In bands 1 through 34 the first column is normally completely black (zeroes); likewise in all the SWIR bands the last column is black. Since the last column for band 77 is completely black (zeroes), the stripes may not be as apparent because the column average variations are small compared to the variation between them and the last column.

- Run ***Auto Destriper***

  ⊕ auto destripe

  ⊕ Destripe Columns

  Number of Iterations 1 *maximum iteration to use should be 2*

  Smoothing Win Size 9 *number of columns, must be an odd number*

  Row Segment Length 60 *number of across-track pixels*

  Number of Similar 15 *number of bands*

  ⊕ Start Destripe. *Note: only bands the band quality of which is set to 'PLOT' will be destriped*

- When complete, use the plane slicer to view the 'before' and 'after' images and also the column average plots for some or all bands (Fig. 24). On the plots, the red curve is the column averages before destriping and the green curve is after destriping. This procedure should have fixed all of the dead pixels and stripes; there should be no reason to manually fix stripes using the ***Unstriping*** module.

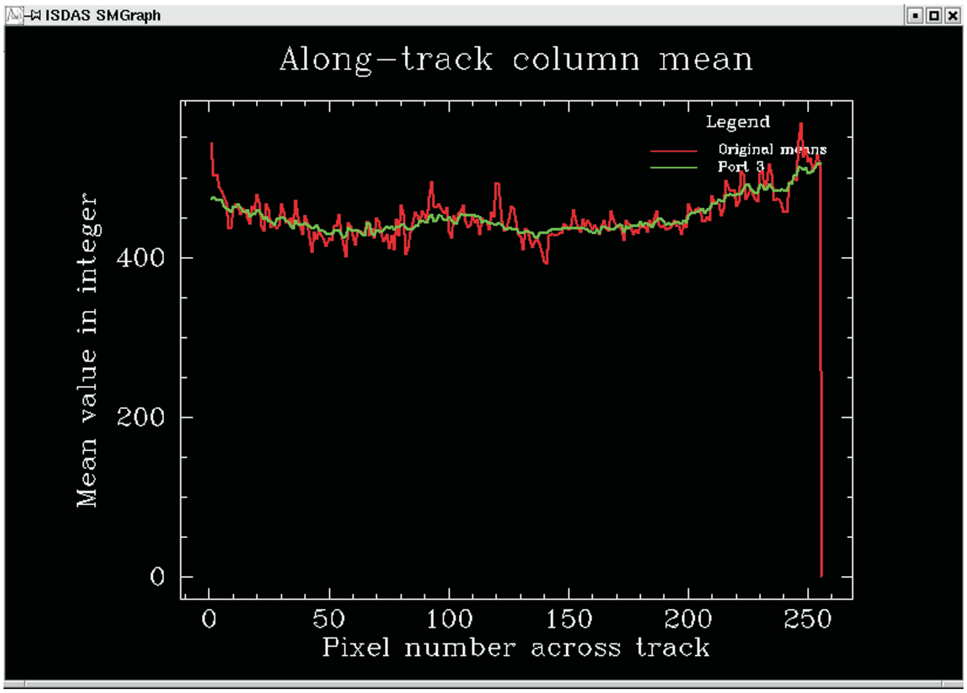


Figure 24. Column average values for a sample data set band before (red) and after (green) implementation of the Auto Destriper module.

- Save the results with ***Data Output***, as **destriped.dts**, **destriped.ssr**, and **destriped.fld**.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

  >clean

2.12. Correct the spatial misalignment between VNIR and SWIR detectors

- At step 2.3, ***Flicker_Tool*** was used to check the spatial alignment of the VNIR to the SWIR. If spatial offsets or rotational misalignment exist, the ***Align Detector*** module will be used to rotate and/or shift the VNIR pixels so they are aligned with the SWIR. Typically (for EO-1 Hyperion data) the VNIR has needed to be rotated approximately 0.222º counter-clockwise. The user may use this value or estimate the amount of rotation and/or shift needed.
- Connect the network as in Figure 25.

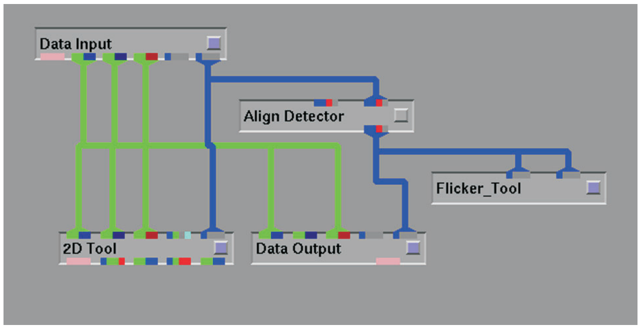


Figure 25. Network used to evaluate potential spatial misalignment between detectors within a sensor.

- Use ***Data Input*** to read in **destriped.dts**.
- Using the ***2D Tool***, view a few bands to verify that the destriping step worked properly.
- Estimate the amount of misalignment using the ***Align Detector*** module. If this amount is already known, then skip the estimate step and go to 'Apply the Correction'.

  ⊕ Align Detector

  ⊕ VNIR

  ⊕ counter-clockwise

  VNIR image bandnumber 50

  SWIR image bandnumber 90

  Maximum x-shift 1

  Maximum y-shift 1

  Maximum angle 1

  ⊕ Estimate

The five best estimates will be displayed in the Linux console window from which ISDAS was started.

- Test the estimates. The test option will only change the VNIR image plane that was used. Typically the first result will be the best at eliminating the misalignment.

  x-shift (as per estimate)

  y-shift (as per estimate)

  rotation angle (as per estimate)

  ⊕ Test only

  Use ***Flicker_Tool*** to see how well the test correction worked on bands 50 and 90.

Once the desired shift has been identified, go to the next step to apply the correction.

- Apply the correction.

  First SWIR band number 71

  x-shift (as per estimate)

  y-shift (as per estimate)

  rotation angle (as per estimate)

  ⊕ Apply correction
- Save the corrected data with ***Data Output*** as **aligned.dts**, **aligned.fld**, and **aligned.ssr**.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

  >clean

2.13. Crop the cube in the line (Y) direction

- Now, identify the ROI to be processed. If processing the entire cube to reflectance, remember that the ***Average_Smooth*** noise removal module may have trouble processing more than 3000 lines (possibly less depending on the Linux system), in which case some lines will have to be cropped (lost) or alternately, if loss of data is to be avoided, the cube can be cropped into two subcubes prior to noise removal and then concatenated into one cube again after noise reduction. Smile detection requires at least 500 lines for optimum performance and noise removal requires a minimum of 1000 lines. It is therefore recommended that a minimum of 1000 lines be kept. The more lines that are kept, the better the noise removal will work, but it will also take longer. If not already done, the line numbers outlining the ROI to be processed can be identified using the cursor in the ***2D Tool***.
- Read in **aligned.dts** using ***Data Input***.

- At step 2.7, the ***crop field*** module was used to crop a subimage to view in the ***Tiler Tool***. Use the same procedure to crop the data cube, leaving only the desired lines.
- View the output in the ***2D Tool*** to verify the results of cropping.
- Save the output as **cropY.dts**, **cropY.fld**, and **cropY.ssr**.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

  >clean

2.14. Spectral space smoothing and noise reduction

- Next, spectral smoothing of the data cube using the ***Average_Smooth*** module is performed to reduce random noise (Khurshid et al., 2006) and increase the signal-to-noise ratio. To do this, edit the **gains.txt** file that was saved with the original data (step 2.2). This file contains a twodimensional array with 256 columns (number of across-track pixels) and 242 rows (number of spectral bands).
- Open the **gains.txt** file in Linux using the text editor (e.g. kwrite).

Note: other text editors may be used, but some, such as emacs, may not recognize the End of Line or Tab characters and hence, not display the data in their proper columns.

  >kwrite **gains.txt**

  Select and delete the first five header lines at the top that begin with semicolons (;).

  Delete the blank line that remains at the top.

  Put cursor on the last line of text in the file.

  Hit 'End' key to go to end of the line, then hit the down arrow.

  If the cursor moves down to a blank line, go back up to the end of last text line.

  'Delete' all blanks until the blank line is gone (i.e. cursor can not move down)

  Hit 'Enter' to put in a carriage return.

  Save as **gains_edit.txt** and exit.

Note: make sure there is one and only one blank line (carriage return) after the last data line, otherwise an error message will appear while reading the gains_edit.txt file into Average_Smooth saying that the sensor and DN (digital number) to radiance factor frame (rad factor frame) are incompatible.

- Create the network shown in Figure 26.

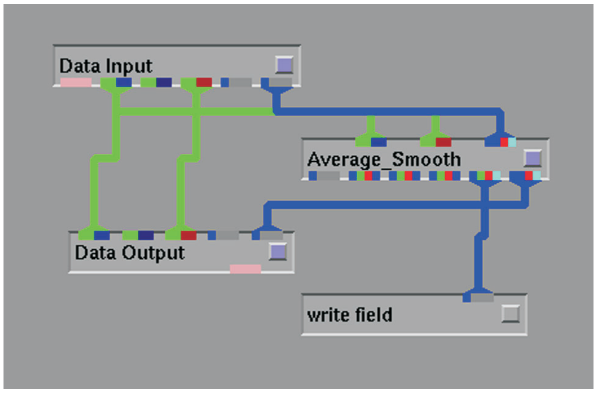


Figure 26. Network used to evaluate and adjust for random noise within a data set.

- Read in **cropY.dts** using ***Data Input***. The module will automatically start to gather statistics.
- Perform noise reduction.

  ⊕ Generate Noise Cube

  ⊕ Read DN to Rad Factor Frame ⇒ **gains_edit.txt**

  ⊕ Parameters for Noise Cube

  number of detectors 2

  detector number 1

  start band 1

  end band 70

  charge conversion factor 78

  noise floor in dn 1.44

  ⊕ normalize spectral cube

  ⊕ don't transform

Note: do not hit the 'Generate' button yet! Enter the second detector information first.

  detector number 2

  start band 71

  end band 242

  charge conversion factor 44

  noise floor in dn 3.4

  ⊕ normalize spectral cube

  ⊕ don't transform

  ⊕ Generate

Note: the entered parameters will come up on the Linux console window.

- When this has completed save the noise cube.

⊕ Save Noise Cube ⇒ **noise_cube.fld**

- Use the noise cube, just generated, to remove noise.

  ⊕ Execute smoothing

  number of regions 1

  region number 1

  start wavelength (*verify that the value is correct*)

  stop wavelength (*verify that the value is correct*)

  Max Numb of Similars 100

  Number of Sigma 1

  ⊕ No PostSearching

  ⊕ Execute

Note: if the module finds a pixel for which there are no 'similar' spectra (for a given sigma value), then a warning message will appear. Ignore this warning and close the warning window. By increasing the number of sigma, the chance of finding similar spectra is increased, but if the value is too high, then spectra that are not similar will be mistakenly selected.

- Check the results. The average number image will appear in the image viewer (Fig. 27). Pixels with no similar pixels, spectrally speaking, will appear dark blue, whereas pixels with 100 similar pixels (the maximum) will appear as red. The majority of the scene should be red, orange, and/or yellow, with the exception of water bodies and shorelines, which will be blue. If there is a lot of blue on land then the noise reduction was not successful and it should be run again with the Number of Sigma increased to 1.5. In EO-1 Hyperion data, there are some anomalous pixels that have spikes at certain wavelengths. These will also appear blue because they have no similar pixels.

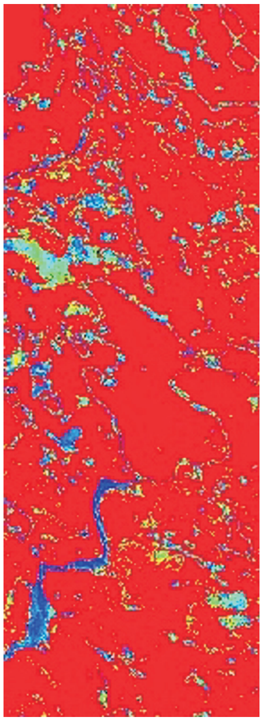


Figure 27. Results of determining pixels with similar spectral radiance subject to random noise limits. Red indicates a significant number of similar pixels, whereas blue indicates few similar pixels.

- Save a screen grab of the Average Number Image.

  > import **avg_number_image.jpg**

- Save the smoothed cube using the ***Data Output*** module as **avg_smooth.dts**, **avg_smooth.ssr**, and **avg_smooth.fld**.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.
- Save the average number image using ***write field***.

  ⊕ write field ⇒ **avg_num_img.fld**

  Using the ***2D Tool*** and ***1D Tool***, view some spectra before and after noise removal to see what effect it had. With EO-1 Hyperion data view bands 77 and 220 specifically.

  >clean

2.15. Crop the cube in the band (Z) direction

- To speed up processing and reduce file sizes, remove all zero bands at the start of the cube (bands 1–7), at the end of the cube (bands 221–242), and in the overlap region (bands 58–76). Since the ***Smile Detector*** module can not handle the SKIP band flag, remove the last two good bands of the VNIR (bands 56, 57). In total 50 bands will be removed, leaving 192 bands in the resultant data cube.
- Create the network shown in Figure 28.
- Read in **avg_smooth.dts** using ***Data Input***.
- Use ***crop field*** to crop bands 1–7, 56–76, and 221–242.

  ⊕ crop field

  ⊕ z (remove band)

  ⊕ band number

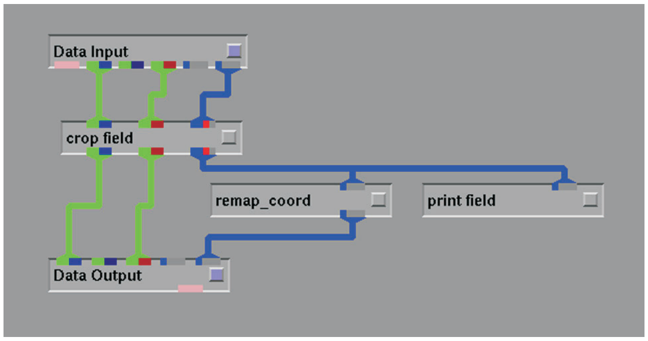


Figure 28. Network used to crop a data set, removing data not to be further analyzed.

from band 1

to band 7

⊕ add to list

from band 56

to band 76

⊕ add to list

from band 221

to band 242

⊕ add to list

⊕ crop cube now

- Use ***print field*** to verify that the Z dimension is 192.
- Use ***remap_coord*** to edit the data cube header so that the z co-ordinates start counting at band 1 and not band 8 as the result of the cropping just performed.

⊕ remap coord

dim 3 start 1

dim 3 end 192

⊕ apply

- Save the cropped cube with ***data output*** as **cropZ.dts**, **cropZ.fld**, and **cropZ.ssr**

Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

2.16. Estimation of the water-vapour range

Smile detection (step 2.20) requires the generation of an atmospheric correction LUT (look-up table) (Staenz and Williams, 1997) calculated at every fourth pixel across the swath (85 pixel locations for EO-1 Hyperion scenes). For nominal atmospheric correction, values are calculated at only five pixels across the swath. An increase in the number of pixels greatly increases the size of the LUT generation time. One of the dimensions in the LUT is water vapour. If the water-vapour range for the scene is known, the time required to generate the large LUT can be greatly reduced. In this step, a small LUT shall be quickly created, calculated at only five pixels across the swath and then a water-vapour map created to get a rough idea of the water vapour range in the scene, prior to creation of the large LUT.

- Create the network as shown in Figure 29.
- Use ***Data Input*** to read in **cropZ.dts**.
- Use the ***2D Tool*** and ***1D Tool*** to view some bands and spectra in order to check all of the preprocessing steps up to this point.

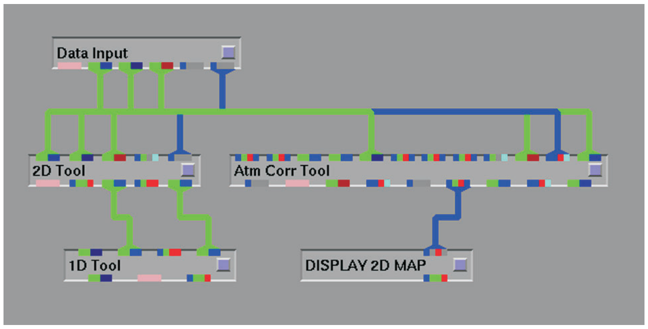


Figure 29. Network used to atmospherically correct a data cube, while simultaneously deriving and displaying atmospheric water vapour.

- Determine the scene elevation using a digital elevation map (DEM) or topographic map. Use this to calculate the mean elevation of the scene or ROI(s). The other option is to read in and use a DEM that has been co-registered to the data cube. Very rarely will a DEM be necessary as most scenes do not have a significant amount of relief and a single average value will suffice.
- Determine the horizontal visibility from in situ measurements, from local weather station data, or use a default value (23 km). For scenes in Canada, archived weather data can be found at (http://climate.weatheroffice.ec.gc.ca /advanceSearch/searchHistoricData_e.html; accessed January 15, 2007) For world weather, try the Aerocan network or http://www.wunderground.com/; accessed January 15, 2007).
- Create the small LUT in the ***Atm Corr Tool***.

⊕ Atm Create Raw LUT

⊕ Radiative Transfer Code ⇒ Modtran4v2

Special controls for Modtran

⊕ Specify

CO2 mixing ratio

⊕ Use 1999 recommended value

⊕ Atmosphere

⊕ Haze ⇒ Continental Rural

⊕ Specify horizontal visibility

start visibility (km) 23

⊕ Atmospheric Gases ⇒ Midlatitude Summer

Water vapor column

⊕ Specify

# of water vapor values 16

Start water vapor (g/cm2) 0

End water vapor (g/cm2) 4

Note: choose a combination of values so that the 'Delta water vapour' (step size) is 0.25. If the scene is over a rainforest, the 'End water vapour' value may need to be increased to 5.

⊕ Solar Irradiance

*Note: verify that the latitude and longitude, time, and date correspond with the information in the *.msn file. The easiest way to do this is in the Data Input module; click on the 'Show Mission' button.*

⊕ Image Coordinates

Image pixel (x) coordinates ⊕ Specify

# of pixel (X) values 5

Terrain elevation setup ⊕ Specify

Start terrain elevation (km) 0.5

Wavelength limits

⊕ Specify limits

Start wavelength (nm) *subtract 25 nm from the wavelength of the first band*

Stop wavelength (nm) *add 25 nm to the wavelength of the last band*

⊕ Viewing Parameters

Note: verify that the altitude, swath angles, and heading are correct.

⊕ Control

⊕ Save parameters **small_LUT.acp**

⊕ Execute

- When it has completed, save the raw and convolved LUTs.

OPTIONAL Save Current Raw LUT

⊕ New File **raw_LUT_small**

⊕ Atm Convolve LUT

⊕ atm convolve LUT

OPTIONAL Save Current Conv. LUT

⊕ New File **conv_LUT_small**

- Create a water-vapour map, using both the 940 nm and 1130 nm water vapour absorption features.

⊕ Atm Transform Cube

⊕ estimate H20 map

Start Wavelength 850

Stop Wavelength 1250

Pixel Step 5

Average Size 3 ⊕ execute

- Save the water-vapour map.

⊕ Write water vapor map

⊕ New File **wvm_small**

- View the water-vapour map using ***DISPLAY 2D MAP***.

A typical water-vapour map (wvm) should look something like Figure 30. Expect to find very high water-vapour values (red) and very low values (blue) in water bodies and clouds. Disregard these anomalies.

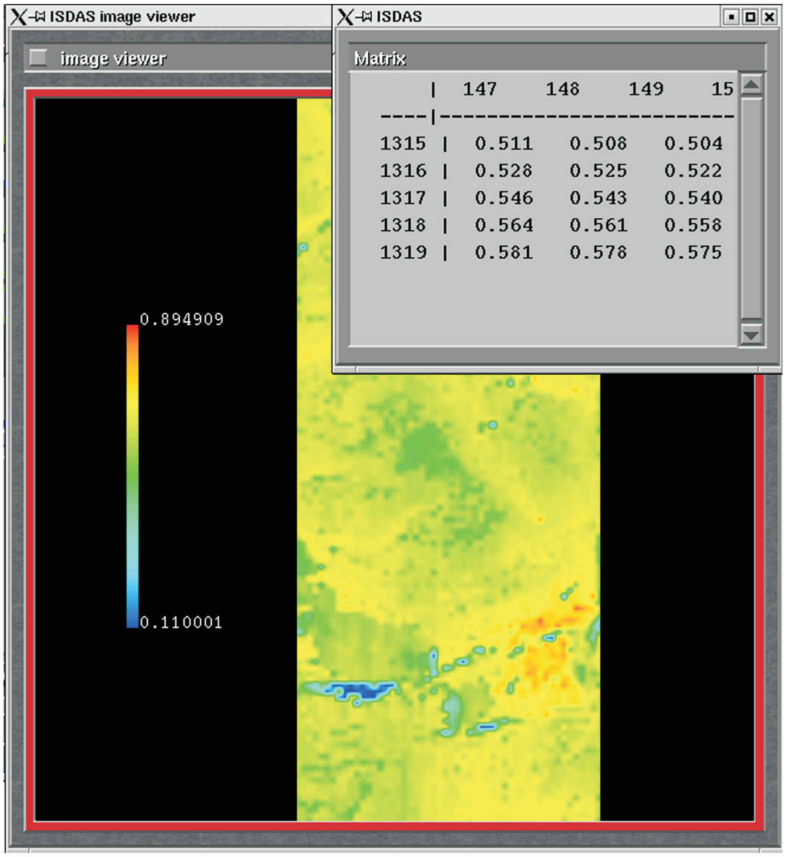
- Make note of the water-vapour range using ***DISPLAY 2D MAP***.

⊕ crosshair

⊕ pick crosshair

⊕ show matrix

Now use the cursor to click on various points in the image. The values will be shown in the Matrix window. If this window is hidden, double click on the 'Show Matrix' button. There will usually be some very low pixel values in the water bodies and a few very high pixel values, usually in the last



```
     |  147    148    149    15
-----|-------------------------
1315 |  0.511   0.508   0.504
1316 |  0.528   0.525   0.522
1317 |  0.546   0.543   0.540
1318 |  0.564   0.561   0.558
1319 |  0.581   0.578   0.575
```

Figure 30. Example atmospheric water-vapour map derived from an EO-1 Hyperion data set.

column. Ignore these when determining the range of values. If the values range between 0.9 and 1.6 for example, then use a range of 0.5 to 2 when creating the large LUT in step 2.17.

Notes: While the water-vapour map is being generated, there are usually a few pixels outside of the water-vapour range that was specified when the LUT was created. These pixels are usually located in water bodies or clouds and are not of much concern. When the water-vapour map has been generated the statistics are reported in the 'Estimate H2O' map console window as well as in an automatically generated file, estimate_wv_map_error.txt, which also lists the locations of the offending pixels. Verify that the location of pixels with values outside of the LUT water-vapour range corresponds to the location of water bodies, clouds, or the last column in the scene. If there are many pixels outside of the water-vapour range and they do not correspond to a water body, then the LUT should be regenerated with an increased water-vapour range and recreate the water-vapour map.

A number of intermediate files are automatically created during the creation of the LUT. These files, mc.dat, tape, MODTRAN4v2.BIN, and atm_create_lut.log are created in the directory from which ISDAS was started. For this small LUT these files can be deleted.*

>clean

2.17. Create large LUT

- Next, the LUT to be used in smile detection and atmospheric correction will be generated. This is referred to as the 'large' LUT because it is calculated at every fourth pixel across the swath as previously discussed.
- Create the network shown in Figure 31.
- Use ***Data Input*** to read in **cropZ.dts**.
- Read in the atmospheric correction parameter (***.acp**) file that was created in step 2.16.

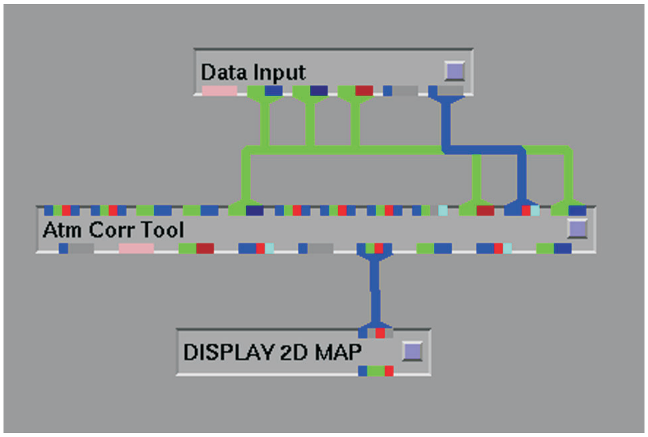


Figure 31. Network used to atmospherically correct a data cube.

⊕ Atm Create Raw LUT

⊕ Control

⊕ Read parameters ⇒ **small_LUT.acp**

⊕ Atmosphere

⊕ Atmospheric Gases

Water vapour column

⊕ Specify

Note: enter the water-vapour range determined in step 2.16, and enter a number of water-vapour values so that the 'Delta water vapour' = 0.25.

⊕ Image Coordinates

Image pixel (x) coordinates

⊕ Specify

#of pixel (x) values 64 *Note: the delta pixel should be approximately every fourth pixel*

⊕ Control

⊕ Save parameters ⇒ **large_LUT.acp**

⊕ Execute *Note: this step may take more than 12 hours to run.*

- Save the raw and convolved LUTs as **raw_LUT_large** and **conv_LUT_large**

>clean

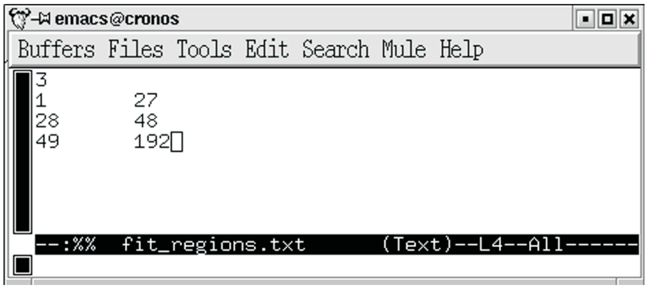
2.18. Edit the gains file

Using a text editor edit the gains file, **gains_edited.txt**, to reflect the cropping done in the band (Z) direction in step 2.15. Start deleting the last lines, first 221–242, then 56–76, then 1–7. Save this new file as **gains_cropZ.txt**. This file will be used to calculate new 'radiometric' gains and offsets during smile detection (step 2.20).

2.19. Create a feature file and a fit regions file

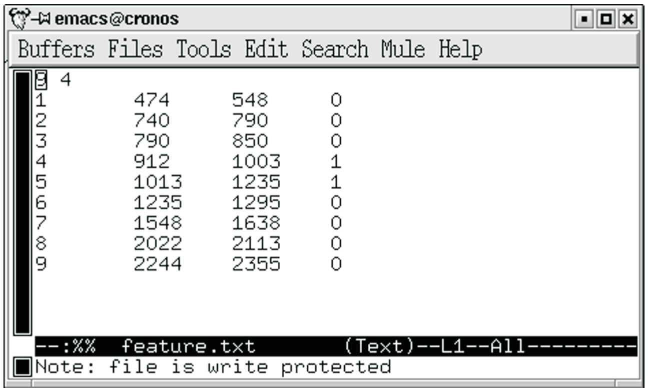
- These files are needed for the smile detection step 2.20.
- Using a text editor, create a fit regions file (Fig. 32) and a feature file (Fig. 33) and save them as **fit_regions.txt** and **feature.txt**.
- The **fit_regions.txt** file is used to split the bands into three regions (2 VNIR and 1 SWIR). Because of the nature of the detectors, the ***Smile Detector*** module will apply a different fit for each region.

In the feature file, the first line contains the number of features to use for smile detection 9 and the second number is the number of columns 4 . The two wavelength regions, containing



```
3
1         27
28        48
49        192
--:%%  fit_regions.txt        (Text)--L4--All------
```

Figure 32. A file used to distinguish the two detector components in an EO-1 Hyperion data set during smile detection.



```
9 4
1         474     548     0
2         740     790     0
3         790     850     0
4         912     1003    1
5         1013    1235    1
6         1235    1295    0
7         1548    1638    0
8         2022    2113    0
9         2244    2355    0
--:%%  feature.txt        (Text)--L1--All---------
Note: file is write protected
```

Figure 33. A file used to identify the band locations of atmospheric absorption features which contribute to the radiance observed in an EO-1 Hyperion data set.

the 940 nm and 1130 nm water-absorption features, have a ‘1’ in the fourth column. The ‘1’ means that gain and offset calculation is turned on for this feature.

2.20. Smile detection

- Create the network as shown in Figure 34.
- Use ***Data Input*** to read in **cropZ.dts**.
- Use the ***Smile Detector*** module to detect the Spectral Line Curvature (Neville et al., 2003)

  ⊕ Setup Parameters

  ⊕ Read in Feature File ⇒ **feature.txt** *Calculations start running automatically*

  ⊕ Wavelength and Bandwidth Range

  Start shift (nm) -4

  End shift (nm) 4

  Shift step (nm) 0.2

  Start percentage -20

  End percentage 20

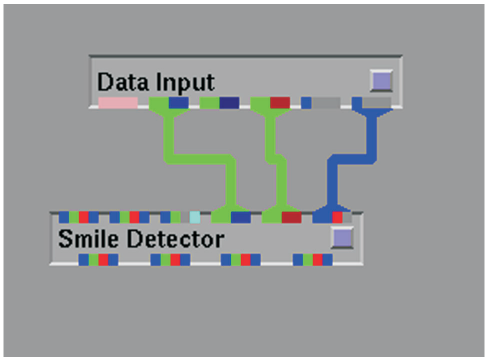


Figure 34. Network used to implement smile detection.

  Changing step 5

  ⊕ Band Direction Polynomial Fit ⇒ **fit_regions.txt**

  ⊕ Piecewise Fit *Note: for sensors other than EO-1 Hyperion, polynomial may apply.*

  ⊕ Read DN to Radiance Gains Frame ⇒ **gains_cropZ.txt**

Note: gains should display on the Linux console window. This step can be skipped if a gain and/or offset detection is not desired.

  ⊕ Atmospheric Correction

  ⊕ Read Raw LUT ⇒ **raw_LUT_large** *Note: the water-vapour estimation will start automatically at this point. Just let it finish.*

  ⊕ WL Range to Estimate Water Vapor

  Start Wavelength 1000

  End Wavelength 1250

  ⊕ Start Smile Detection

  Number of iteration 5

  ⊕ Iterative Execute

- Save the results to disk before viewing the results to protect against data loss due to an ISDAS crash. Use the nine buttons to save all nine files as **WL_cali.fld**, **WL_cali.txt**, **BW_cali.fld**, **BW_cali.txt** **DN_gain.fld**, **DN gain.txt**, **DN_offset.fld**, **DN_offset.txt**, and **GO_corrected.fld**.
- *Note: only save this last file if gain and offset detection was turned on for any of the features.*
- Copy **cropZ.dts** to **GO_corrected.dts** and **cropZ.ssr** to **GO_corrected.ssr** and then edit them appropriately i.e. The ***.dts** file points to **GO_corrected.fld**.
- Visually check the results.

Arrange the four SMGraph windows so that they can be viewed simultaneously. SMGraph <1> and SMGraph <4> are related as are SMGraph <2> and SMGraph <3> so arrange them logically as in Figure 35.

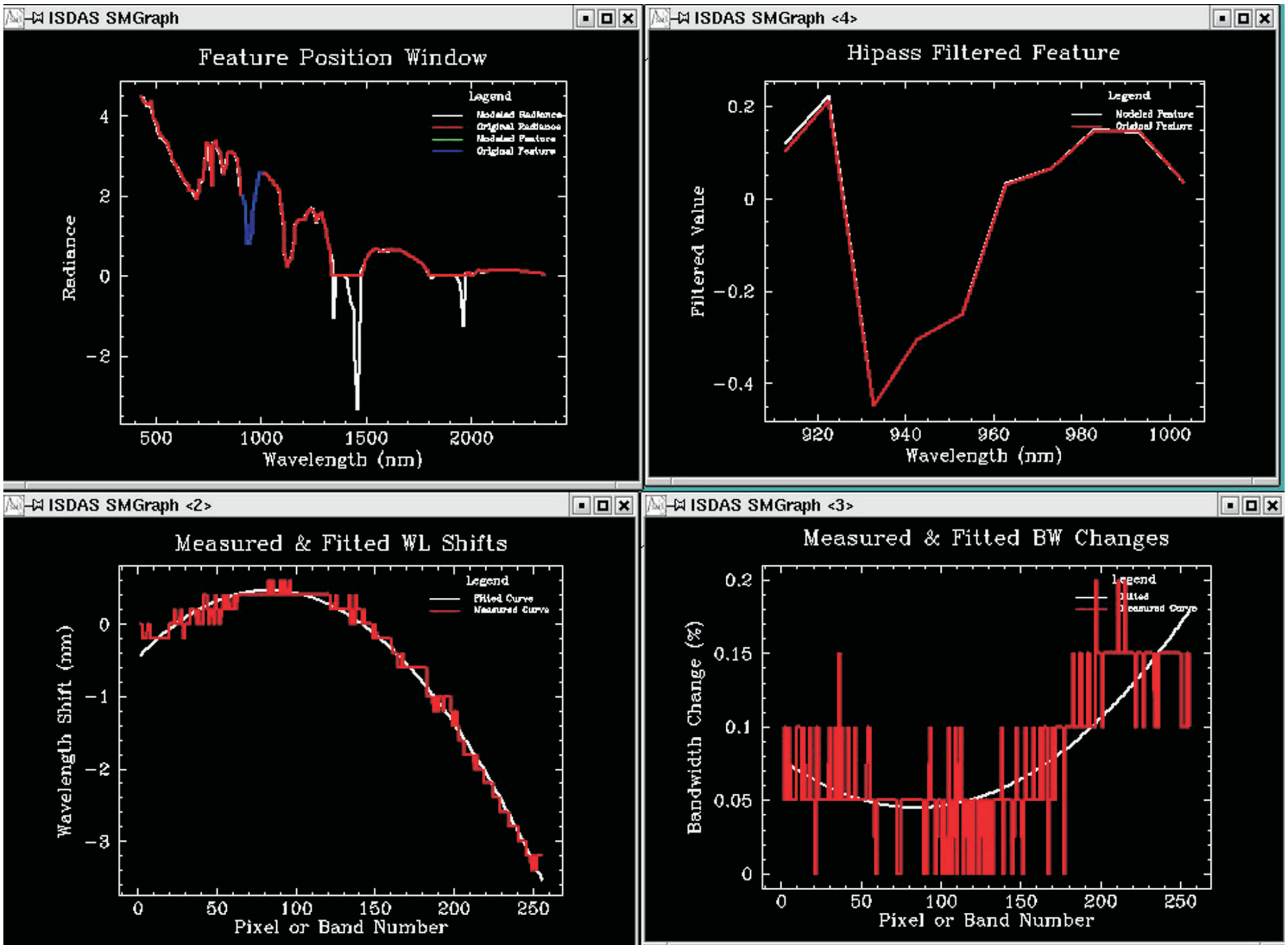


Figure 35. Example of output plots resulting from running smile detection. Original data is displayed in red, modelled data used to evaluate the impact of spectral smile is in white, and highlighted features are displayed in blue.

⊕ Visually Check Results

⊕ Check correlation results

Feature number 1

Pixel position 1

Note: if the contents of the SMGraph windows get erased (totally black) just reselect the 'Feature number'.

The Feature Position Window (SMGraph <1>) shows two spectra. The original radiance spectrum is shown in red. This is the average spectra for column 1. The modelled radiance spectrum is shown in white. This is a result of taking the original radiance, converting to reflectance, smoothing it, and applying all combinations or wavelength and bandwidth changes and then choosing the best fit.

The blue highlighted part of the spectra indicates the feature that has been selected to display; initially this is feature 1. Visually verify that the blue covers the entire desired absorption feature as well as its 'shoulders'. Typically there should be one spectral band on either side of each feature shoulder.

The 'Hipass Filtered Feature Window' (SMGraph <4>) shows the original and modelled radiance in red and white, respectively, but it only shows the selected feature wavelength range. The shape will not appear exactly like it does in the 'Feature Position Window' because both spectra have been run through a high-pass filter. Ideally, both spectra should be very similar. If they are not, verify that the correct gains are in the ***.dts** file and also consider running gain and offset detection for that feature.

For each of the nine features, check a few pixel positions and make sure that everything looks okay.

⊕ Check WL & BW tables

⊕ Band direction

Pixel Position 1

SMGraph <2> and <3> show the measured and fitted wavelength (WL) shift and bandwidth (BW) change as a function of wavelength. By selecting different pixels, it can be observed how these change across the swath. Do not spend too much time reviewing these band direction graphs, however, as the pixel direction graphs give a better indication of the performance of the smile detection.

⊕ Pixel direction

Feature Number 1

SMGraph <2> and <3> show the measured and fitted WL shift and BW change as a function of pixel location. By selecting different pixels, observe how these change across the swath. Red represents measured BW changes and WL shifts and white represents the curve of best fit. As the shape that the WL shift typically takes is that of a parabola it is referred to as the 'smile', although often it is an inverted smile or frown.

As each feature is selected and viewed, look to see if, for any feature, the curves exceed the maximum or minimum WL shift (-4 to 4) or BW changes (-0.2 to 0.2). If they do, then this module should be rerun with expanded limits. The WL smile curves for a given detector should be the same general shape (i.e. concave or convex); however they may differ between detectors. If the first feature (720 nm) is not the same shape as the other three features of the VNIR detector, do not worry as it is a weak feature. The four strongest and most significant features are the 760 nm, 820nm, and 940 nm in the VNIR and the 1130 nm in the SWIR.

>clean

2.21. Atmospheric correction

- Create the network shown in Figure 36.

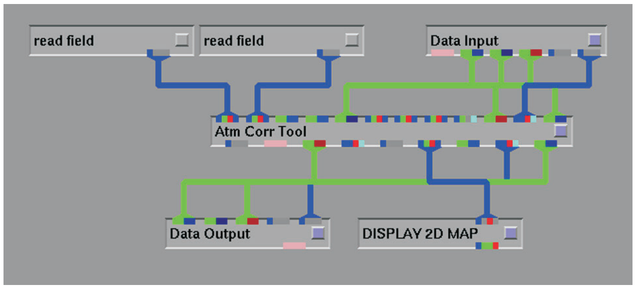


Figure 36. Network used to implement atmospheric correction incorporating results from the smile detection.

- Use ***Data Input*** to read in **GO_corrected.dts**.
- Using ***read field***, read the bandwidth calibration field, **BW_cali.fld** into the first from left input port of ***Atm Corr Tool***.
- Using ***read field***, read the wavelength calibration field, **WL_cali.fld** into the second from left input port of ***Atm Corr Tool***.
- Using the ***Atm Corr Tool*** module, read in the **raw_LUT_large** and convolve it to the new bandwidth and wavelength calibration tables.

⊕ Atm Convolve LUT

⊕ atm convolve LUT

OVERRIDE Current Input LUT with…⇒ **raw_LUT_large**

- Create a water-vapour map, using both the 940 nm and 1130 nm water-absorption features.

⊕ Atm Transform Cube

⊕ estimate H20 map

Start Wavelength 850

Stop Wavelength 1250

Pixel Step 5

Average Size 3

⊕ Execute

- Save the water-vapour map.

⊕ Write water vapor map

⊕ New File **wvm**

- View the water-vapour map using ***DISPLAY 2D MAP***.

It should resemble the water-vapour map created at step 2.16. If there are pixels for which the water-vapour values are outside of the range used to create the LUT, then visually verify that they are in clouds or water bodies.

- Atmospherically correct the radiance data cube to produce reflectance.

⊕ Atm Transform Cube

⊕ atm transform cube

⊕ Execute

Note: in the Linux console window the band numbers should appear, increasing from 0 to 191 as each band is processed.

- Save the reflectance cube as **b4_smilecor_refl.dts**, **b4_smilecor_refl fld**, and **b4_smilecor_refl ssr** using ***Data Output***.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

> clean

2.22. Smile correction

- In order to make all pixels in the swath direction of a certain band correspond to the same wavelength centre and bandwidth, a spectral resampling process is required. This step corrects the wavelength shifts and bandwidth changes for each pixel across the swath.
- Create the network shown in Figure 37.

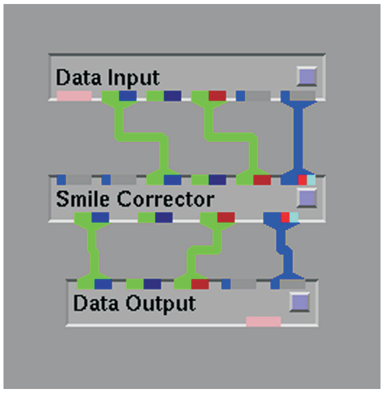


Figure 37. Network used to resample each column of atmospherically corrected data to provide each individual image band a unique spectral band centre.

- Use ***Data Input*** to read in **b4_smilecor_refl.dts**.
- Read in the ASCII version of the wavelength and bandwidth calibration tables.

  ⊕ Read ASCII Wavelength Cali Table ⇒ **WL_cali.txt**

  ⊕ Read ASCII Bandwidth Cali Table ⇒ **BW_cali.txt**

  ⊕ Apply Cali Table

  ⊕ Execute

- Save the reflectance cube as **refl.dts**, **refl.fld**, and **refl.ssr** using ***Data Output***.
- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.
- Use the ***2D Tool*** and ***1D Tool*** to view some spectra.

The output from smile correction is a ground reflectance data cube that has been corrected for detector misalignment, striping, noise, gain and offset error, wavelength shift, bandwidth change, and atmosphere. The resultant reflectance spectra may still not be perfect; there could still be some noise and unwanted spikes. At this point the user has a number of options.

If the data quality is unacceptable, then the user may redo the atmospheric correction and/or smile detection using different parameters, e.g. the aerosol model.

It may also help to redo noise removal over the 940 nm and 1120 nm water-vapour regions with an increased 'Number of Similars'. This is referred to as strong noise removal.

Another option might be to remove the spikes using the ***Postprocessing*** module (step 3.2).

If the data quality is acceptable, then the data can be saved in a format easily imported into other image processing and GIS software (*see* step 3.1).

Products can also be generated from this reflectance cube. These products may include end-members and resultant fraction maps, land-cover classification maps, normalized difference vegetation index (NDVI) , and biochemical and biophysical indicators such as leaf area index (LAI), chlorophyll, water, or nitrogen content.

3. ADDITIONAL ISDAS PROCESSING TASKS

3.1. Exporting the data cube to RAW format

- In order to export an ISDAS-format radiance or reflectance cube in a format (RAW) that can be imported into other image processing and analysis software such as PCI or ENVI:
- Create the network shown in Figure 38.

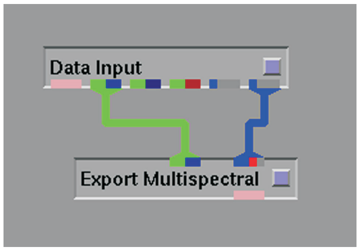


Figure 38. Network used to export a native ISDAS format data set to a raw format that can be imported to other processing or analysis software.

- Read in the desired data cube using ***Data Input***.
- Using the ***Export Multispectral*** module, convert the data type from short (16 bit integer) to float (32 bit real).

  ⊕ Convert data type

  ⊕ Convert short to float

  ⊕ Convert cube now

- Write out the cube

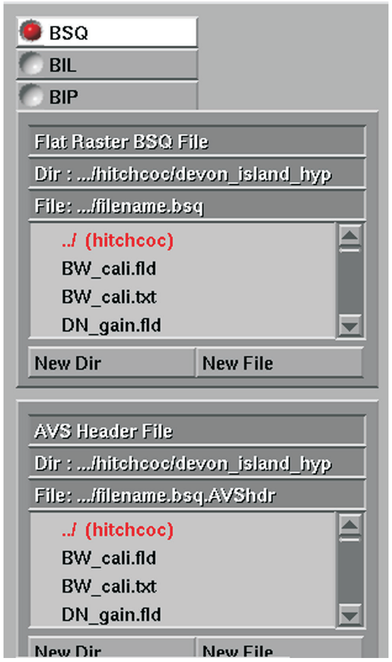
  ⊕ export field

Choose ⊕ BSQ, ⊕ BIL, or ⊕ BIP

Enter the flat raster filename: ⊕ New File **filename**
Note: do not enter a filename extension. It will be added automatically.

If everything is correct at this point, the name of the file should appear in both the ‘Flat Raster File’ and ‘AVS Header File’ windows as shown in Figure 39. If it does not appear in the ‘AVS Header File’ window, then check that the filename (including complete directory pathname) is not too long.

⊕ Execute



● BSQ
○ BIL
○ BIP

Flat Raster BSQ File
Dir : .../hitchcoc/devon_island_hyp
File: .../filename.bsq

../ (hitchcoc)
BW_cali.fld
BW_cali.txt
DN_gain.fld

New Dir | New File

AVS Header File
Dir : .../hitchcoc/devon_island_hyp
File: .../filename.bsq.AVShdr

../ (hitchcoc)
BW_cali.fld
BW_cali.txt
DN_gain.fld

New Dir | New File

Figure 39. Example of an ISDAS dialog window for the Export Multispectral module.

3.2. Postprocessing

- In order to remove unwanted spikes from the reflectance spectra, the ***Postprocessing*** module can be used to automatically determine spectrally flat target pixels in the scene and use these to determine a new gain and offset at each along-track column per band.
- Create the network shown in Figure 40.
- Use ***Data Input*** to read in the reflectance cube. At this point, band 1 of the cube will be displayed in the image viewer and the spectrum at pixel location 1,1 will be displayed in the SMGraph window. The actual reflectance spectrum is shown in blue and the smoothed spectrum, using the default manual smoothing filter is shown in purple.

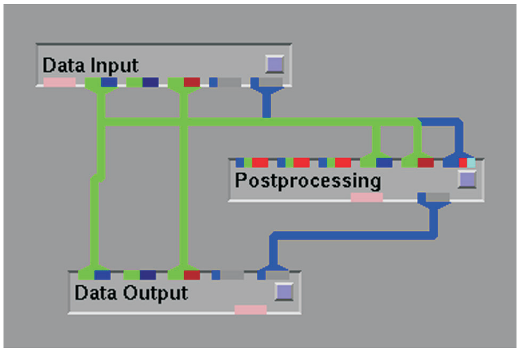


Figure 40. Network used to postprocess an at-surface reflectance data set.

- The ***Postprocessing*** module can be run in two modes. The filter mode involves the user defining a filter which is passed over each spectrum. The controls for this filter are found in the ‘Smoothing options’ window. The other mode involves a flat-target approach. This is the recommended method that will be outlined below as it decreases the chance of removing actual wanted absorption features from the spectra.
- The flat-target method first considers that the EO-1 Hyperion wavelength range is split into three separate regions: VNIR (bands 1–90), SWIR1 (bands 106–136), and SWIR2 (bands 155–192). For each region, VNIR for example, a second-order polynomial is passed through the spectrum for each pixel. The absolute value of the difference between the polynomial and the actual spectrum is calculated at each band and the sum of these differences is recorded for each pixel location in a VNIR difference image. This is repeated for SWIR1 and SWIR2. The mean and standard deviation of each difference image is calculated. The user enters the number of sigma to be used to determine what is considered a spectrally flat target in each region. The flat target pixels are identified for each of the three wavelength regions, resulting in three ROI images. In addition, the chlorophyll absorption integral (CAI) is calculated in order to identify vegetation spectra. This results in a fourth ROI image. The spectra of vegetation pixels will typically be spectrally flat in the SWIR1 and SWIR2 regions, so the unions of the vegetation ROI with the SWIR1 and SWIR2 ROIs are performed. The spectra for each pixel in each of the three ROIs are then smoothed using a Gaussian filter of window size specified by the user. For each column of each band in a particular spectral region, the mean of the original (nonsmoothed) values for all of the pixels in the ROI is calculated. Each pixel with a value higher than the mean is classified as ‘bright’ and each pixel lower is ‘dark’. The mean of both the bright and dark pixels is now calculated. This is now repeated for the smoothed values, resulting in four values (the bright and dark means for both the original and smoothed values) that are then plotted on a graph. A line joining the two points is

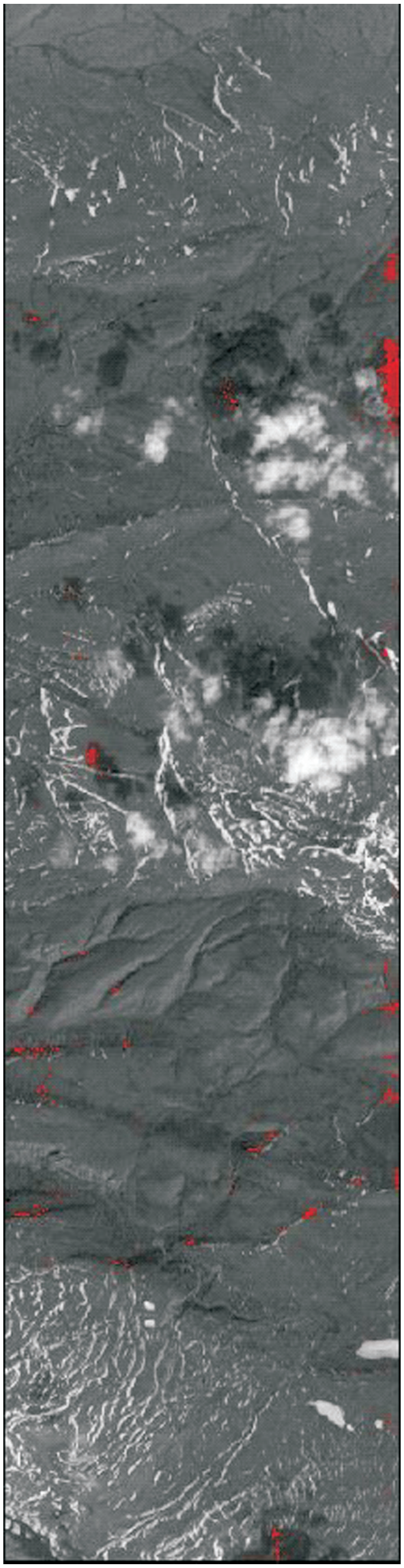


Figure 41. Example of results when postprocessing a data set for **a)** insufficient number of spectrally flat pixels identified;

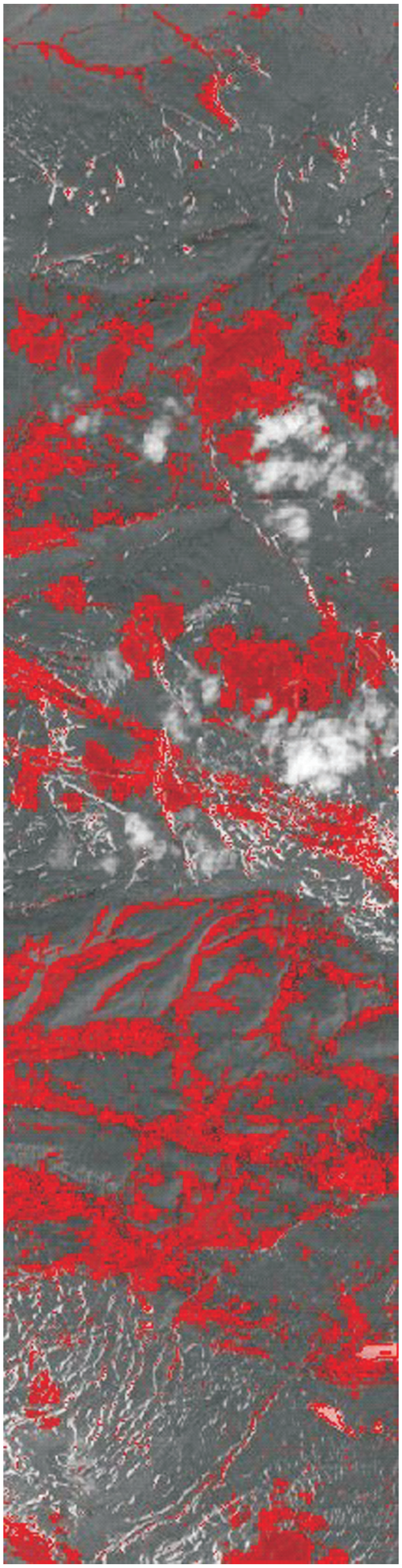


Figure 41. b) sufficient number of spectrally flat pixels identified.

calculated. This line determines the transformation coefficients used to smooth each pixel (not just ROI pixels) along a particular column for a particular band in a particular spectral region.

- Perform an initial flat-target postprocessing run.

  ⊕ Segment Non-feature Targets

  VNIR Window Size 5 *Note: choose an odd number for the window size.*

  VNIR Numb of Sigmas 1

  SWIR1 Window Size 5

  SWIR1 Numb of Sigmas 1

  SWIR2 Window Size 5

  SWIR2 Numb of Sigmas 1

  ⊕ Execute

When finished, the image viewer will, by default, show the vegetation ROIs as red polygons. As well, the result of the smoothing will be shown in purple in the SMGraph.

- Inspect the results to see if the window size and sigma need modifying.

  ⊕ Check the Results

- Click on the VNIR, SWIR1, and SWIR2 buttons to view the ROIs of the flat-target pixels. A poor result (not enough pixels) will look like Figure 41.
- A good result will look like Figure 41b . Typically the problem will be not enough pixels selected and this can usually be remedied by decreasing the number of sigma for the region in question. This should only be done if it is observed that there are spectra in the scene that are spectrally flat in the region of interest and that were not included in the ROIs. To determine this, use the 'pick crosshair' option to probe the image for pixels the spectra of which (in the SMGraph) appear flat in the region of interest.
- For those regions that did have an acceptable number of flat-target pixels chosen, carefully inspect the SMGraph (Fig. 42) and see if the unwanted spikes have been removed (smoothed). If they have not been removed then the window size should be increased to 9, for example. The window size refers to the number of bands, not the number of nanometres.
- Make the appropriate changes to the window sizes and sigmas and rerun ***Postprocessing*** if necessary.
- Save the postprocessed cube using ***Data Output***.

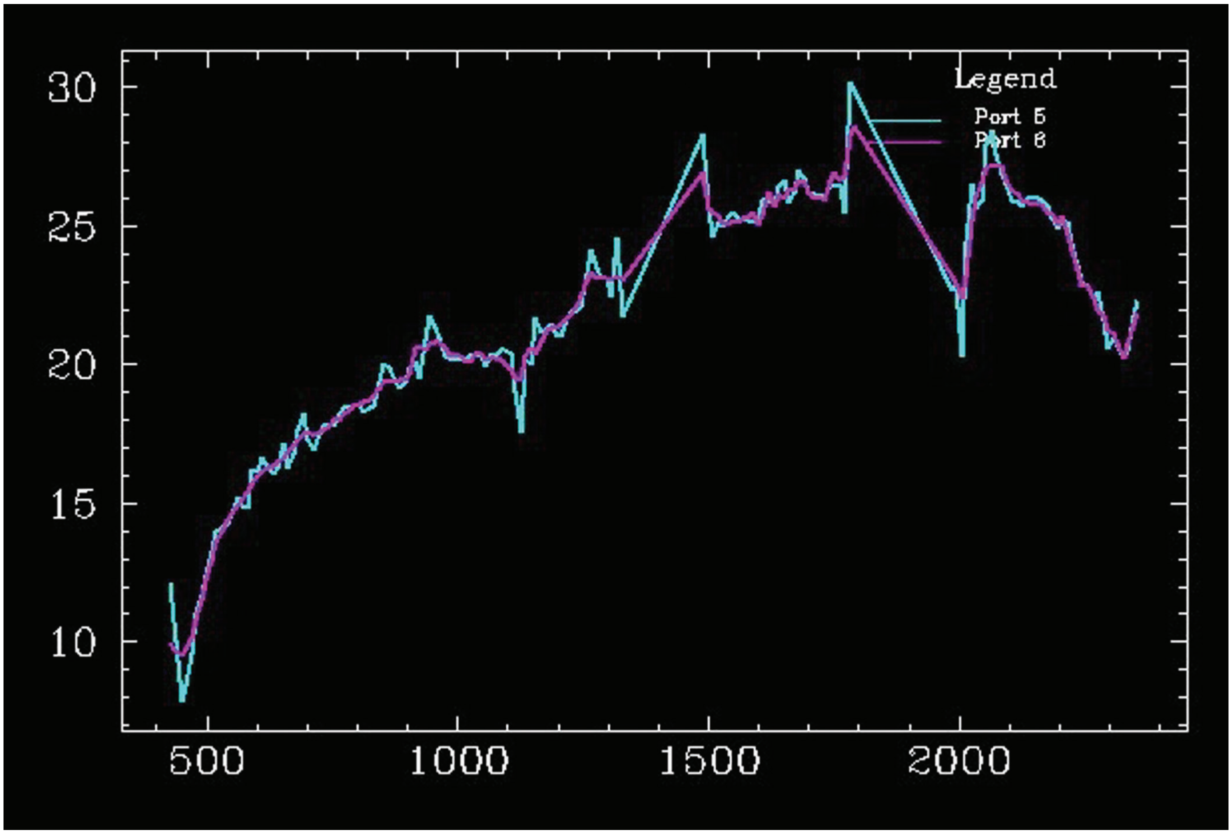


Figure 42. Example plot of original (blue) and postprocessed (red) spectral reflectance.

- Edit the ***.dts** file to remove absolute paths, keeping only the filenames. Edit the ***.ssr** file to remove the Gain and Offset lines.

Use the ***2D Tool*** and ***1D Tool*** to view some spectra.

ACKNOWLEDGMENTS

The authors would like to thank the research groups who provided spaceborne hyperspectral imagery and comments during the research, development, and testing of the several methodologies and algorithms referred to in this document. Thanks specifically to David Goodenough and his team at the Pacific Forestry Centre; Paul Budkewitsch and his team at the Canada Centre for Remote Sensing, Natural Resources Canada; John R. Miller and Baoxin Hu of York University; and Benoit Rivard and Arturo Sanchez-Azofeifa of the University of Alberta. The authors also thank members of the Hyperspectral User Group at the Canada Centre for Remote Sensing who provided significant review, comments, and contributions to this effort.

REFERENCES

Khurshid, S., Staenz, K., Sun, L., Bannari, A., Champagne, C., Hitchcock, R., Neville, R., and White, H.P.
2006: Preprocessing of EO-1 Hyperion data; Canadian Journal of Remote Sensing, v. 32, no. 2, p. 74–83.

Neville, R.A., Sun, L., and Staenz, K.
2003: Detection of spectral line curvature in imaging spectrometer data; *in* Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery IX, (ed.) S.S. Shen and P.E. Lewis; v. 5093, p. 144–154.

Staenz, K. and Williams, D.
1997: Retrieval of surface reflectance from hyperspectral data using a look-up table approach; Canadian Journal of Remote Sensing, v. 23, no. 4, p. 354–368.

Staenz, K., Szeredi, T., and Schwarz, J.
1998: ISDAS - a system for processing/analyzing hyperspectral data; Canadian Journal of Remote Sensing, v. 24, Issue 2, p. 99–113.

Sun, L., Neville, R., and Staenz, K.
in press: Automatic destriping of Hyperion imagery based on spectral moment matching; Canadian Journal for Remote Sensing.

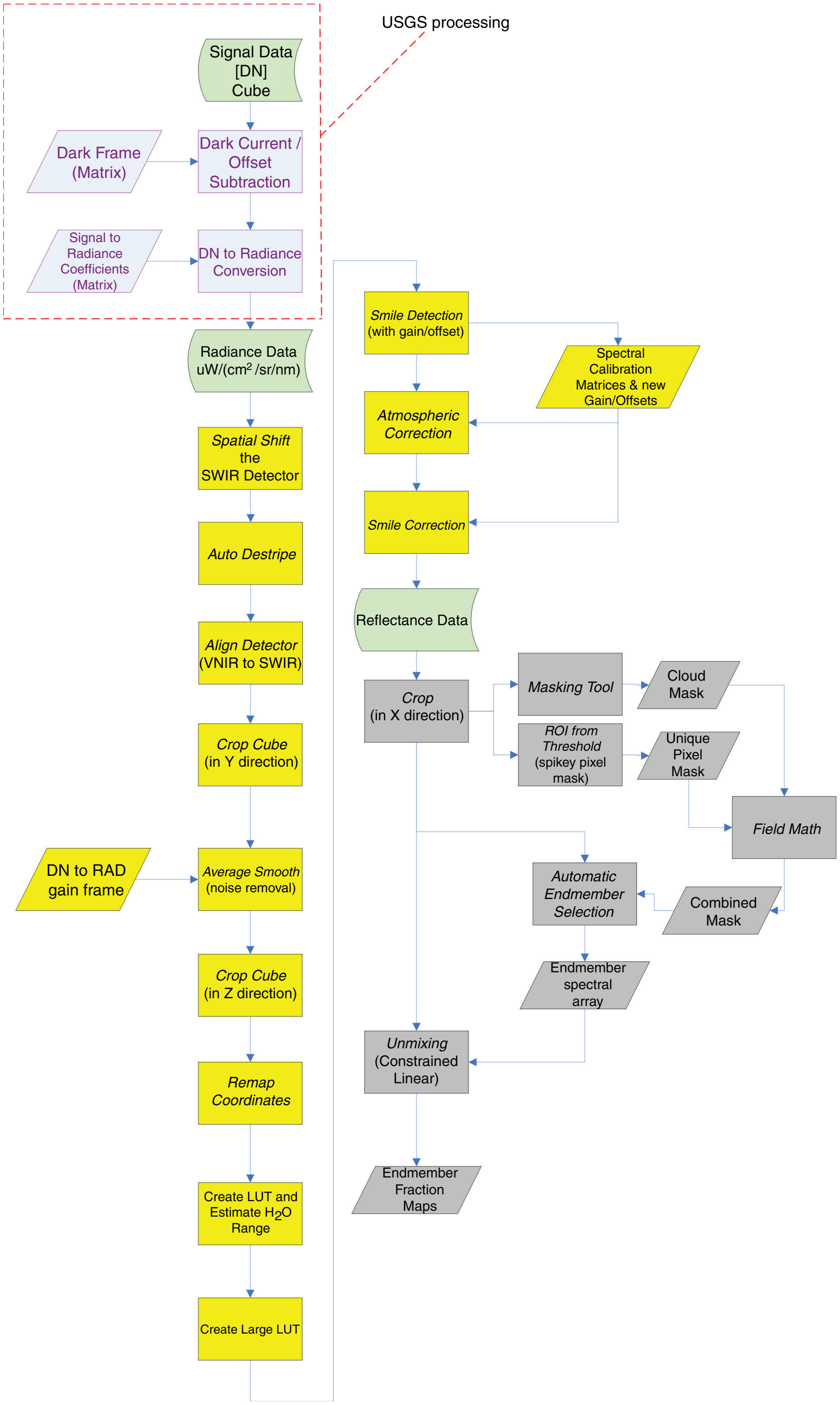
APPENDIX A

Location of ISDAS modules

Module name	Library	Class
1D Plot	isdas_main	Data Output
1D Tool	isdas_main	Data Output
2D Tool	isdas_main	Data Output
Align Detector	isdas_full	Filters
Atm Corr Tool	isdas_main	Filters
Auto Destriper	isdas_main	Filters
Average Smooth	isdas_main	Filters
concatenate cubes	isdas_full	Filters
crop field	isdas_main	Filters
Data Input	isdas_main	Data Input
Data Output	isdas_main	Data Output
DISPLAY 2D MAP	isdas_main	Data Output
edit_band_quality	isdas_full	Filters
Export Multispectral	isdas_main	Data Output
Flicker_Tool	isdas_full	Data Output
Hyperion sensor	isdas_full	Data Input
Import Multispectral	isdas_main	Data Input
Postprocessing	isdas_main	Filters
print field	supported	Data Output
read field	supported	Data Input
read_sensor	isdas_full	Data Input
remap coord	isdas_full	Mappers
show sensor	isdas_full	Data Output
Smile Corrector	isdas_main	Mappers
Smile Detector	isdas_main	Mappers
spatial shift	isdas_full	Filters
Tiler Tool	isdas_main	Mappers
Unstriping	isdas_full	Filters
write field	supported	Data Output
write sensor	isdas_full	Data Output

APPENDIX B

EO-1 HYPERION processing flow chart


USGS processing
Signal Data [DN] Cube
Dark Frame (Matrix)
Dark Current / Offset Subtraction
Signal to Radiance Coefficients (Matrix)
DN to Radiance Conversion
Radiance Data uW/(cm^2 /sr/nm)
Spatial Shift the SWIR Detector
Auto Destripe
Align Detector (VNIR to SWIR)
Crop Cube (in Y direction)
DN to RAD gain frame
Average Smooth (noise removal)
Crop Cube (in Z direction)
Remap Coordinates
Create LUT and Estimate H_2O Range
Create Large LUT
Smile Detection (with gain/offset)
Spectral Calibration Matrices & new Gain/Offsets
Atmospheric Correction
Smile Correction
Reflectance Data
Crop (in X direction)
Masking Tool
Cloud Mask
ROI from Threshold (spikey pixel mask)
Unique Pixel Mask
Field Math
Automatic Endmember Selection
Combined Mask
Endmember spectral array
Unmixing (Constrained Linear)
Endmember Fraction Maps