HiiHAT: The Hyperspectral Image Interactive Holistic Analysis Toolkit

Version 1.03, June 2011

Lead Developer: Brian D. Bue (brian.d.bue@jpl.nasa.gov)
Principal Investigator: David R. Thompson (david.r.thompson@jpl.nasa.gov)

I. Overview

Hyperspectral imagery has provided dramatic new insight into the geology and atmosphere of other planets. However, understanding these images can be quite challenging since scientists can only visualize a small number of bands. The Hyperspectral Image Interactive Holistic Analysis Toolkit (HiiHAT) is an intelligent assistant to help analysts efficiently browse, summarize, and search hyperspectral images. The software consists of a plugin to the IDL/ENVI environment. The algorithms are designed for the special challenges of high dimensional terrestrial and planetary science datasets:

- **High noise levels**: Many of the most interesting planetary science questions involve spectral features at the limits of detectability. We emphasize robust strategies capable of detecting subtle spectral features with high levels of noise.
- **Uncertain constituents**: In many hyperspectral imaging settings, we have very few ground truth samples from the surface. We address this by "unsupervised" analysis that looks for patterns in the observed data itself, and optionally by incorporating domain knowledge in the form of spectral libraries.
- **Fast turnaround time**: Tactical observation planning may require fast decisions, so the analysis procedures aim to be as automated as possible.

The software toolkit includes automatic procedures for image summarization, and techniques to search images for key spectral features.

II. System Requirements

- ITT ENVI+IDL 4.3+ (http://ittvis.com)
- Optional: CRISM Analysis Tool (CAT): required for pre-processing CRISM image formats.
  Available at: http://pds-geosciences.wustl.edu/missions/mro/crism.htm
- Tested on Red Hat Enterprise Linux Client release 5.6 (Tikanga) x86_64, Mac OSX 10.4+ Intel and Windows XP

III. Installation

1. Unzip the outer archive directory, and place it into the save_add directory of your local ENVI installation.
2. Launch ENVI. You should see a new "HiiHAT" menu item appear with several new menu options.
3. You're done! Read further for information on testing, configuration and specific menu options.

Post-Installation Testing

Once you have completed the installation steps, you should run the included unit tests to make sure that HiiHAT is functioning properly on your system. To run the tests, simply type “hiihat_run_tests” at the IDL command prompt (after starting ENVI). This will build and execute the tests in the unit test suite using the (included) MGUnit software (http://mgunit.idldev.com/). When the tests complete, the final line of output should indicate that all of the tests passed. If this is not the case, please send an email to hiihat-dev@jpl.nasa.gov with your system configuration, ENVI/IDL version, which test failed, and a copy of the idl console output (if available).

IV. Configuration File and Menu Options

The HiiHAT plugin includes a configuration file, “hiihat.cfg,” which allows you to customize output parameters, set numerical limits for HiiHAT functions, and configure other HiiHAT-specific details. These parameters are loaded when you when you first execute any of the HiiHAT functions after starting ENVI, or when you choose the “Reload configuration parameters” menu option in the HiiHAT configuration submenu. Additionally, you can save the current set of parameters as defaults, via the “Save configuration as default” menu item. Saved parameters will be used if no matching parameter is given in the configuration file. The list of configuration options is given in Table 1, below:

Table 1: Configuration Parameters

Parameter	Allowed Values (0=Disable, 1=Enable)	Description
verbose	0, 1	Enable/disable verbose console output (default=0)
debug	0, 1	Enable/disable debug console output (default=0)
gui_status	0, 1	Enable status updates in GUI form for non-essential output messages, instead of console output (default=0)
finite_max	[0,max_float]	Maximum value to be considered “finite” for pixel-based comparisons (default=10000)
robust_means	0, 1	Enable/disable robust mean calculation. Ignores spectra with infinite values or values of zero in all bands (default=0).
nfindr_iterations	[1,1000]	Number of iterations to run the N-FINDR endmember detection algorithm (default=10)
check_endmember_redundancy	0, 1	Enable/disable checking for nearly identical potential endmembers before performing endmember detection (warning: may be slow if the set of potential endmembers is large) (default=0)
backup_rois	0, 1	Saves rois calculated before endmember detection (default=1)

V. HiiHAT Functions and Experiments

The HiiHAT plugin appears as a new menu in the ENVI toolbar. This section explains each of the menu options in detail. For additional documentation on specific HiiHAT functions and their parameters, see the doc/idldoc/index.html file in your HiiHAT installation directory. It contains IDLdoc-based (http://idldoc.idldev.com/) documentation for all provided HiiHAT functions.

Required Data for HiiHAT Experiments

In order to run the experiments provided in this document, you need the following datasets (provided in the envidata directory that comes with the IDL/ENVI installation set, or alternatively, can be downloaded from the ittvis website at the following address:

http://www.ittvis.com/portals/0/tutorials/envi/20_Advanced_Hyperspectra.zip

File	Description
cup95eff.int	Atmospherically calibrated, EFFORT-polished AVIRIS imagery from the Cuprite, NV test site. Subset to the 1.9-2.4 micron range. Integer valued.
cup95eff.hdr	Header file for cup95eff.img image.

Preprocess Image

Many of the analysis routines in the toolkit require a clean reflectance spectrum with consistent formatting conventions (i.e. missing data set to zero). We have designed preprocessing routines that will convert several data products of interest to the HiiHAT standard. These include the images specified in Table 2: Accepted Image formats. If you wish to process an image type that is unspecified, you can select “Generic” image to apply the HiiHAT preprocessing functions to your image.

Our preprocessing function serves four roles:

I. **Selection of spectral bands of interest on for known image types** (given in Table 2): These bands are selected based on expert knowledge of the most relevant bands for each image type.

II. **Median filtering in the spectral domain**: median filtering is desirable as it reduces shot noise and column striping for noisy images. We provide an option to apply a spectral median filter, with a radius specified as a number of spectral bands in this function. We recommend a value of 2-3 for noisy images (e.g. CRISM), or 0 for less noisy datasets (e.g. AVIRIS).

III. **Spectrum normalization**: “Raw” pixel values in many images are subject to differences in illumination effects. One way to compensate for this is to divide each spectrum by its Euclidean norm. Additionally, in some cases, it may be desirable to use a function other than the Euclidean norm to mitigate scaling issues between spectra in a given image.

IV. **Mean division**: it is often advantageous to “divide out” the mean of the image rows or spectral bands, depending on the characteristics of the imagery you analyze, or type of analysis to be performed.

We suggest you perform the “preprocess” step before continuing to the superpixel processing options. The preprocessing options are selected automatically in the “Summary Classification” routine.

Table 2: Accepted Image Formats

Type	Description
Generic (default)	Generic image type (default), no band selection or filtering performed Selected band range: all bands.
CRISM FRT (CAT-processed)*	CAT Preprocessed CRISM Full Resolution Targeted Image Selected band range: 1.067-2.6 μm
CRISM MRDR (CAT-processed)*	CAT Preprocessed CRISM Multispectral Reduced Data Record Image Selected band range: 1.067-2.6 μm
CRISM MSW (CAT-processed)*	CAT Preprocessed CRISM Multispectral Window Image Selected band range: all bands
CRISM MSP (CAT-processed)*	CAT Preprocessed CRISM Multispectral Survey Image Selected band range: all bands
M3 L1B RDN (raw)	M3 Level1B Radiance Image Selected band range: 0.75-2.0 μm, lowpass filter with kernel size 3
USGS AVIRIS reflectance	Airborne Visible/Infrared Imaging Spectrometer Image Selected band range: 1.067-2.6 μm
EO-1 Hyperion 10-band	10 Band Hyperion Multispectral Image Selected band range: all bands
EO-1 Hyperion 12-band	12 Band Hyperion Multispectral Image Selected band range: all bands

*requires the CRISM CAT toolbox available at http://pds-geosciences.wustl.edu/missions/mro/crism.htm

Example 1: Preprocessing AVIRIS data

Here, we will preprocess the Cuprite95 AVIRIS image using HiiHAT’s Preprocess Image function.

1. Open the cup95eff.int image in ENVI.
2. Because this image is integer-valued, we first need to convert it to floating point before preprocessing it. To do this, select “Basic Tools” -> “Stretch Data” and select the cup95eff.int image. You will need to select a range of values to stretch the current integer reflectances. Choose the (arbitrary) range [0,1000], and save the image as cup95eff.img.
3. From the HiiHAT menu, select “Preprocess Image.” You can select an output file for the preprocessed image, or select cancel to store the preprocessed output image in memory. For large files, we advise outputting the image to file as to not exhaust the memory on your machine. In this case, set the output image value to cup95eff_m2_l2n.img.
4. After choosing the output method, the Preprocessing Options dialog box will appear. Here, we will perform a +/- 2 band spectral median filtering operation on the image, and normalize each spectrum in the image by the Euclidean norm.
5. Since this image is a (spectral) subset consisting of 50 of the original 240 AVIRIS bands, select “Generic Image” (rather than USGS AVIRIS, which assumes all 240 bands are included) for the image type and click ok.
6. Figure 1 shows false color plots of the original (left) and l2 normalized/median filtered image (right), and Figure 2 shows an example pixel at the center position of the red box in Figure 1 in the original (left) and preprocessed image (right).

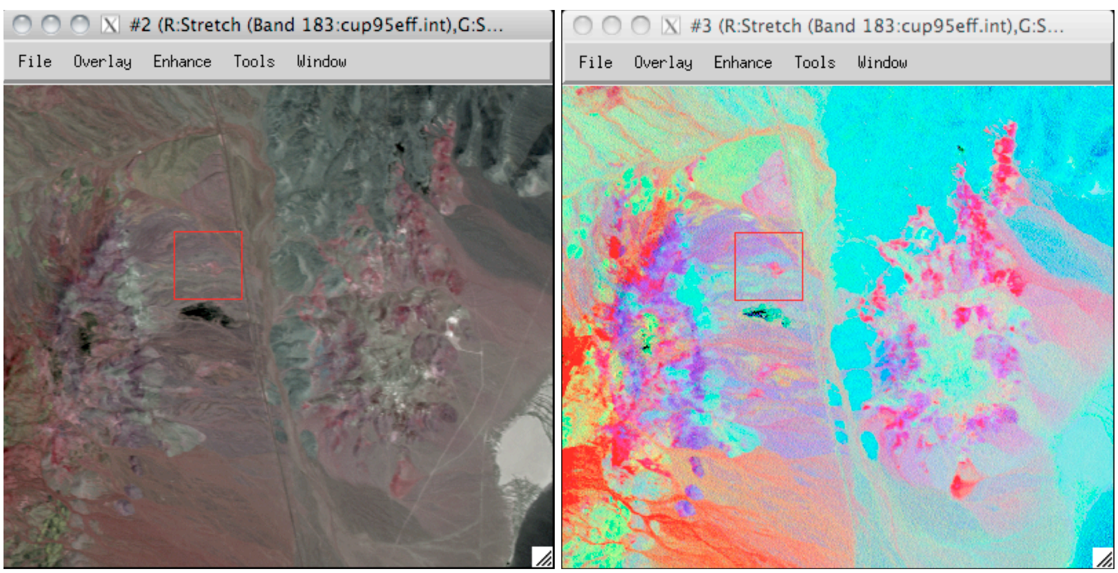

Figure 1: False color image of cup95eff.img before (left) and after (right) preprocessing. The same bands were used to generate both images, but colors are exaggerated (as expected) after dividing each spectrum by the Euclidean norm (right).

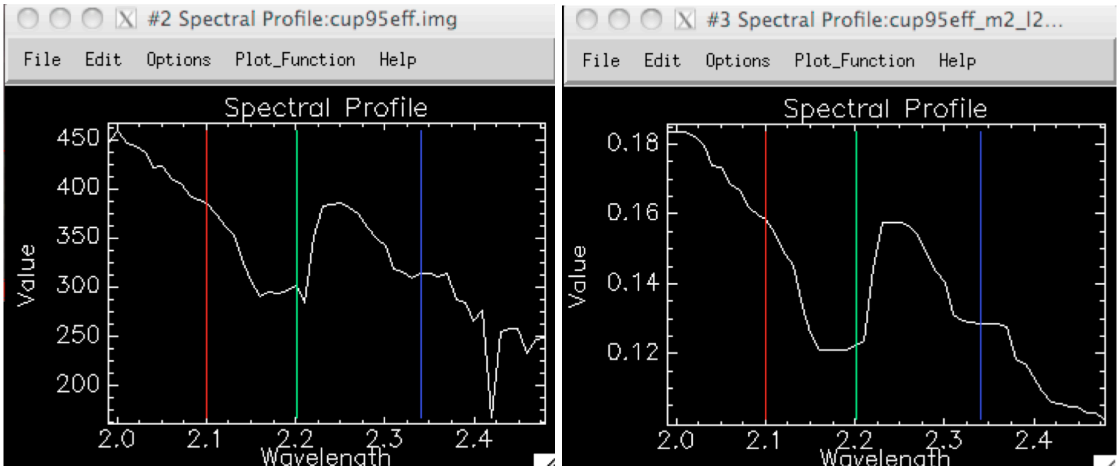

Figure 2: cup95eff.img spectra before (left) and after (right) preprocessing with width 2 median filter and Euclidean normalization. Red, green and blue vertical lines indicate the selected bands in the false color images in Figure 2.

Superpixel Utilities

Superpixel Segmentation

The "**Superpixel Segmentation**" option performs a graph-based segmentation of an image, using the algorithm described by Felzenszwalb et al. [Felzenszwalb et al 2004]. Superpixels are an oversegmentation of the scene. They can be an effective noise reduction strategy and are used here to improve the result of endmember extraction on noisy images.

The segmentation method depends on three main parameters: First, the 'C' parameter biases the size (and quantity) of the superpixels, with larger values favoring smaller superpixel sizes. The second parameter, "minimum size," sets a threshold where segments with fewer pixels than the threshold will be merged into neighboring superpixels. The last option allows for different metrics for calculating spectral distances to generate the segmentation. Ideal values for these parameters depend on the data you analyze – but the default parameters are, in general, a good starting point for most datasets.

Example 2: Segmenting the 1995 Cuprite Image

Here, we will use HiiHAT to segment the preprocessed cup95eff_m2_l2n.img file we created in the previous section.

1. Select the "Superpixel Segmentation" item in the HiiHAT menu. You will be prompted for an input image file and an output image file. Select the cup95eff_m2_l2n.img file as input, and save the output as cup95eff_m2_l2n_seg_c0.001.img.
2. You will next be prompted for the three parameters "C", "minimum size", and "distance metric." Set the C parameter to 0.001, the minimum size value to 20, and choose "Euclidean" for the distance metric.
3. Segmentation results for the cup95eff_m2_l2n.img file with the suggested parameters are shown in Figure 3 (left). For contrast, we provide a segmentation map with the C parameter set to 0.1. As you can see, larger C values produce larger segments, while smaller values produce a larger quantity of small segments.

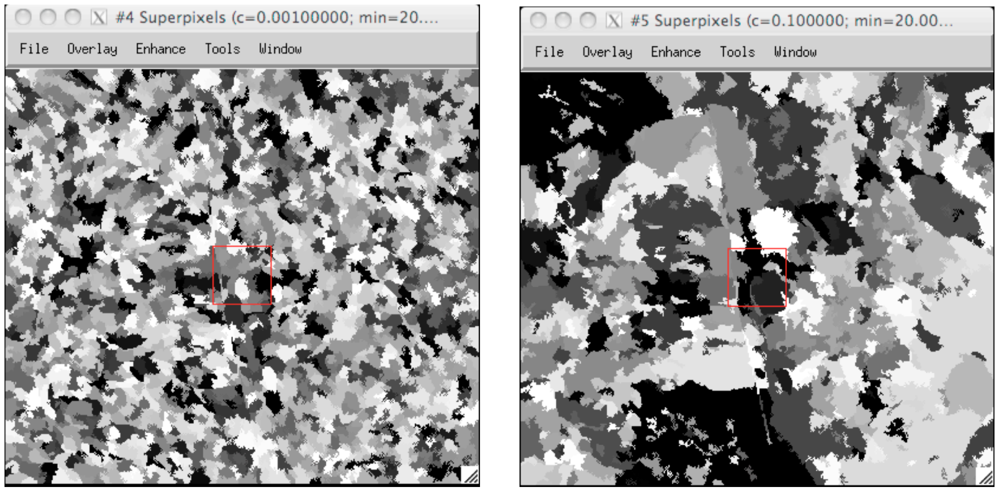

Figure 2: Influence of C on pixel size. Segmentations of cup95eff_m2_l2n.img with c=0.001 (left) and c=0.1 (right). Larger c values produce segmentations with fewer and larger superpixels.

Superpixel Endmember Detection

The "**Superpixel Endmember Detection**" option takes an image and a superpixel segmentation of the image, extracting the mean spectrum from each superpixel and performing endmember detection using either the SMACC [Gruniger et al. 2004] or the N-FINDR [Winter, 1999] method on the superpixel mean spectra. The function will output both a spectral library file containing endmember spectra, and also a Region of Interest (ROI) file showing the locations of the endmember superpixels in the image. You must choose the number of endmembers to extract and the method to use in order to extract them. If you choose to use SMACC, you can specify a "coalesce value" which coalesces any endmembers that are within the specified spectral angle mapper threshold into a single endmember. The most extreme spectra are identified and used to represent the entire coalesced group of endmembers. A value of zero disables this option.

Warning: it is possible that the endmember detection routines will fail if some of the pixels/superpixels are approximately identical to one another. If the endmember detection routine fails or crashes, you can enable the "check_redundant_endmembers" option in the hiihat.cfg file to check if any identical spectra in the set of endmember candidates. This process is slow, however, and should only be enabled if endmember detection initially fails.

Example 3: Detecting endmembers in the Cuprite image

Here, we will use HiiHAT to detect a set of endmembers using the preprocessed cup95eff_m2_l2n.img file (created in Example 1), and the segmentation data we created in the previous section.

1. In the HiiHAT menu, select "Superpixel Endmember Detection."
2. Choose the cup95eff_m2_l2n.img file as the input image, and select cup95eff_m2_l2n_seg_c0.001.img as the input segmentation.
3. Save the output file as cup95eff_m2_l2n_seg_c0.001_endm.sli and the ROIs as cup95eff_m2_l2n_seg_c0.001_endm.roi.
4. You will be prompted for the number of endmembers to detect, and the endmember detection method. Choose 10 for the number of endmembers, and select SMACC as the endmember detection method. Select zero for SMACC coalesce value.
   - After setting these parameters, HiiHAT will select the most spectrally distinct endmembers from the mean spectra of the superpixels. The resulting endmembers are shown in Figure 4, and can be viewed with the "Spectral Library Viewer" in the ENVI "Spectral" menu.
5. For comparison, we try endmember detection on the raw image pixels, rather than the superpixels. To do this, in the ENVI menu, select "Spectral", "SMACC Endmember Extraction." Give 10 as the number of endmembers,

and set the output file as cup95eff_m2_l2n_pix_endm.sli, and leave the remaining parameters as default.

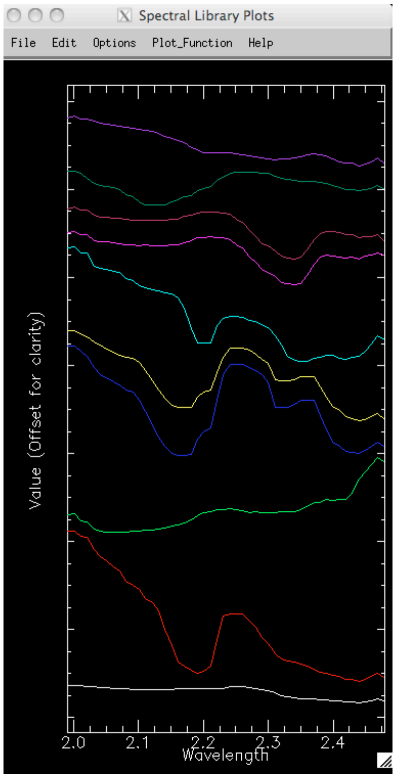

Figure 3: SMACC superpixel endmembers calculated from superpixel segmentation of cup95eff_m2_l2n.img (minsize=20, c=0.0001). Noise artifacts are significantly reduced in comparison to Figure 4. (Note that the order and the colors in this figure do not correspond to those in Figure 4).

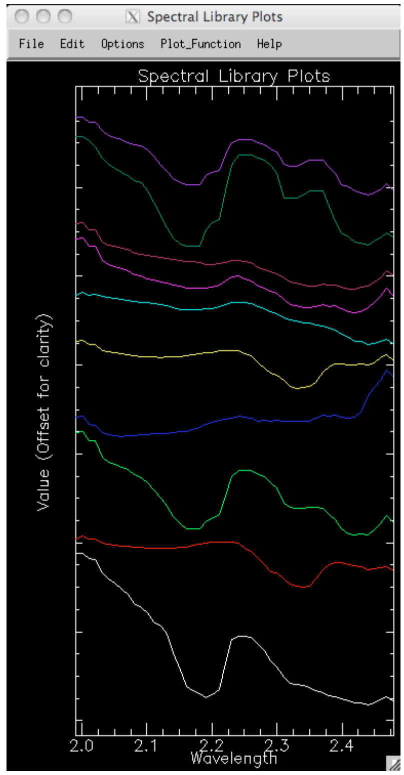

Figure 4: N-FINDR superpixel endmembers calculated from superpixel segmentation of cup95eff_m2_l2n.img (minsize=20, c=0.0001). The set of endmembers produced by N-FINDR are similar to SMACC, but not identical (e.g. cyan, red, magenta), due to differences between the two algorithms.

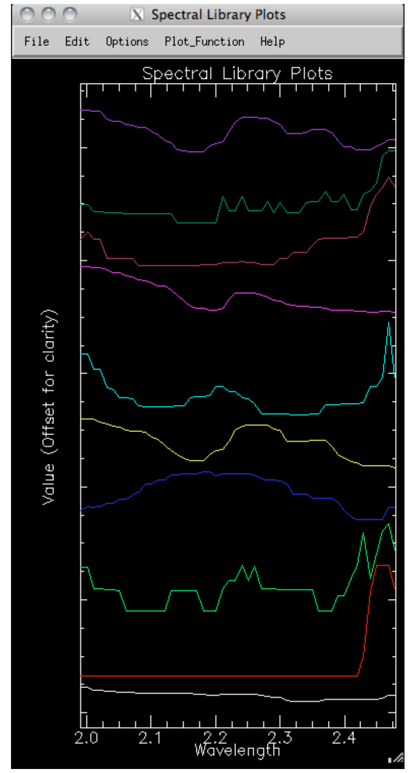

Figure 5: SMACC pixel endmembers calculated from cup855ee_m2_l2n.img raw pixels. While some of the endmembers may be mineralogically accurate (e.g., purple, magenta, yellow, blue), several endmembers are likely noisy outliers (e.g. dark green, green, red, pink).

Superpixel Neutral Region

A common strategy for enhancing interesting mineral features in reflectance spectra to divide spectra of interest by a so-called neutral spectrum. This also can often reduce or eliminate systematic distortions in image spectra caused by sensor noise, varying illumination conditions, or atmospheric effects. The "**Superpixel Neutral Region**" option will automatically detect a neutral spectrum using the method described in [Mandrake et al. 2010].

Example 4: Detecting and Dividing Out Superpixel Neutral Regions

1. Here, we use the neutral region detection function to detect and "divide out" the neutral region for a given image.
2. In the HiiHAT menu, select the "Superpixel Neutral Region" menu option.
3. Select the cup95eff_m2_l2n.img file as the input image, and cup95eff_m2_l2n_seg_c0.0001.img as the segmentation image.
4. Save the output of the neutral region function as cup95eff_m2_l2n_seg_c0.0001_neut.sli
5. The Neutral Region detection function returns a spectral library that can now be viewed with the "Spectral Library Viewer" in the ENVI "Spectral" menu, along with a copy of the image where each pixel in the image is divided by

the detected region.

6. Open the “cup95eff_m2_l2n_seg_c0.001_neut.sli” in the Spectral Library viewer and click on the “neutral spectrum” to view the detected neutral region.

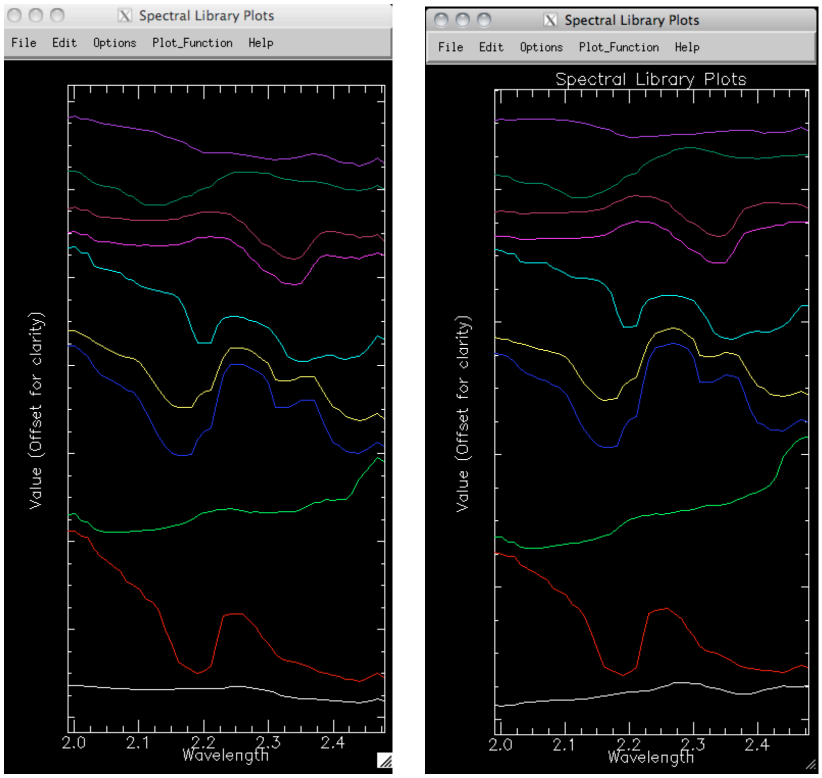

Figure 7: SMACC superpixel endmembers before (left) and after (right) dividing out the neutral region detected using the procedure above. While the differences between these spectra may appear small, they can make a significant difference when comparisons between spectra from different images (or other spectral libraries) are necessary.

Superpixel Mean Image

The “**Superpixel Mean Image**” option will allow you to combine an image and an existing segmentation of that image into a “mean image,” where each pixel in the original image is replaced by the mean of the segment to which it belongs.

Example 5: Creating a Mean Image for an Existing Segmentation Image.

1. Here, we will create a mean image for cup95eff_m2_l2n.img using the cup95eff_m2_l2n_seg_c0.001.img segmentation.
2. In the HiiHAT menu, select the “Calculate Mean Image” menu option.
3. You will be prompted for an input image and a segmentation image. Select the cup95eff_m2_l2n.img and cup95eff_m2_l2n_seg_c0.001.img, respectively. Save the mean image as cup95eff_m2_l2n_seg_c0.001_mean.img.
4. The resulting mean image (Figure 10, top right) for the segmentation image (Figure 10, top left) is calculated and saved.

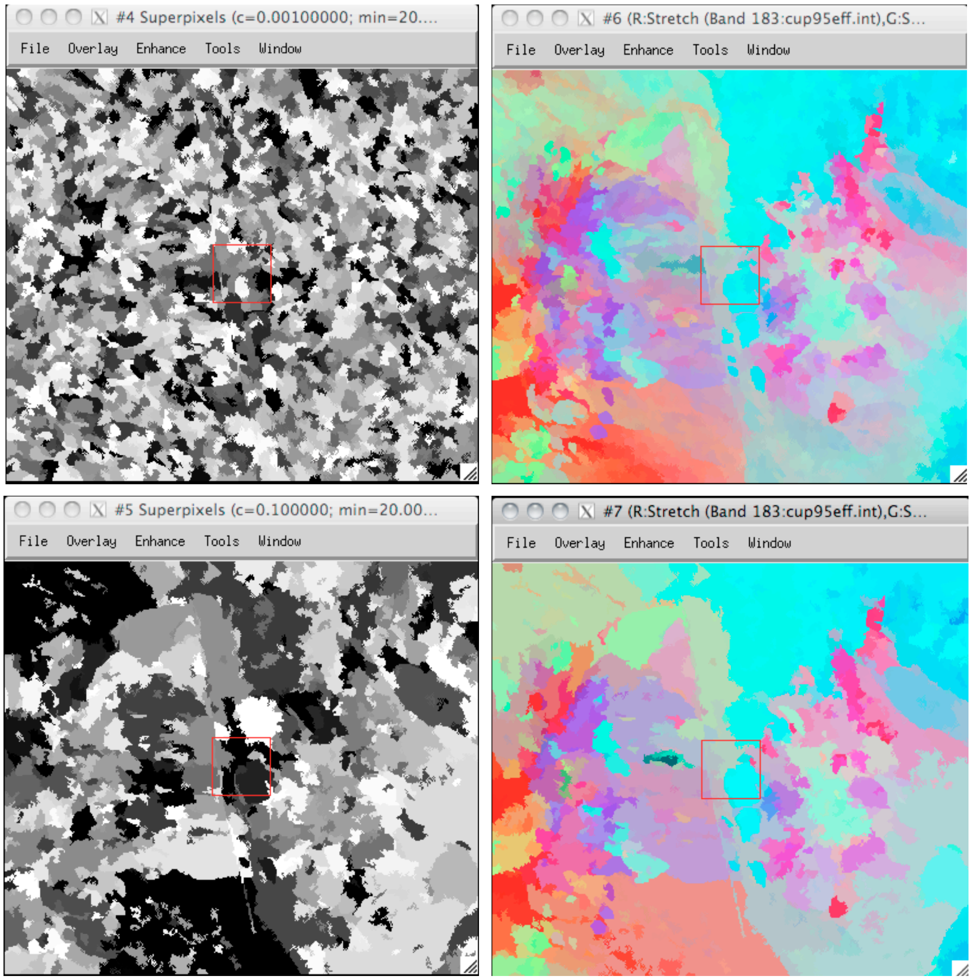

Figure 8: cup95eff_m2_l2n.img superpixel segmentations and corresponding mean images for C=0.001 (top) and C=0.1 (bottom)

Summary Classification

This routine automates a simple, automated classification of the input image. A series of dialog boxes will guide you through the processes of preprocessing, superpixel segmentation, endmember extraction and classification. For further information on the steps performed in by this function, see the previous sections. For further information on image classification methods, please refer to the ENVI documentation.

Example: Classifying cup95eff.img

1. From the HiiHAT menu, select the “Summary Classification” item.
2. You will be prompted for a classification method and an output file for the classification. Select SAM (Spectral Angle Mapper) for the classification method, select the cup95eff.img as your input file, and name the output file cup95eff_m2_l2n_sam.img.
3. You will next be prompted for a set of parameters for the preprocessing, segmentation and endmember detection procedures, given below. For additional information on these parameters, see the previous sections.
   - Preprocessing parameters: image type=Generic, median filter width=2, normalization=Euclidean, and Mean Division=None.
   - Segmentation parameters: set “min sizeONE BAG ” to 20, “C” to 0.0001 and “distance metric” to “Euclidean.”

- Endmember detection parameters: select 10 endmembers and use the SMACC endmember detection routine with coalesce threshold 0.
- Classification parameters: select SAM as the classifier, and choose cup95eff_class.img as the class image, and cup95eff_class_rule.img as the output rule image.

4. After these parameters have been defined, HiiHAT will preprocess the image, calculate the superpixel segmentation (Figure 3, left), detect the 10 endmembers from the superpixels using SMACC (Figure 4), and will then use those endmembers to classify the image using the Spectral Angle Mapper, assigning pixels to classes according to the closest endmember spectrum. The resulting classification is shown in Figure 9: Summary Classification of cup95eff.img.
5. Note that every pixel in the scene is classified using this method. In some cases this is not desirable, as many of the pixels may be very different from the endmembers, but are still assigned to the nearest endmember "class" by the selected classifier. We can eliminate these poorly classified pixels in post processing by selecting thresholds for the classes provided in the rule image using the "Rule Image Classifier" function in the ENVI "Classification" menu.

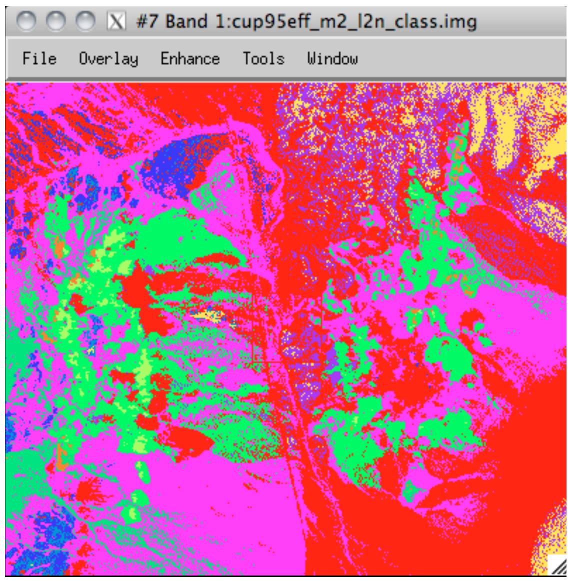

Figure 9: 10-class summary classification of cup95eff.img. Pixels are assigned to the class label of their most similar endmember.

Metric Learning

Here, we will use a set of ROIs defining a set of classes - in this case, the "class" ROIs produced by the endmember detection function - to transform our data in a way such that the distances between classes are spread out, while similar pixels are moved closer to one another using Linear Discriminant Analysis as described in [Bue et al. 2011]

Example: Learning a metric for the cup95eff_m2_l2n.img and its endmembers.

1. If it is not open already, open the "cup95eff_m2_l2n.img" and double click one of the image bands to load it to a new display.
2. In the image display, select the "Tools," "Region of Interest," and "ROI Tool" function.
3. In the ROI Tool, select any currently displayed ROIs with the "Select All" button and delete them with the "Delete ROI" button.
4. Select the "Restore ROIs" option from the "File" menu in the ROI Tool and load the "cup95eff_m2_l2n_seg_c0.001_endm.roi" file to load the endmember rois.
5. From the HiiHAT menu, select the "Learn Metric from ROIs" item.
6. You will be prompted for the image to transform, the number of samples per class, the "regularization amount"

and an output file for the transformed image. Setting the regularization amount value to 100% will produce a more statistically accurate transformation but may take a long time to process. Setting it to 0% will process the image quickly, but the transform may be less accurate as a result.

7. Select the cup95eff_m2_l2n.img file, choose 100 samples per class, 50% for the regularization amount, and name the output file “cup95eff_m2_l2n_lda.img.”
8. The output image will consist of (number of classes)-1 bands, with each band corresponding to an output space in which the classes are better separable. The first band is the best separating space, while the last band is the least separable w.r.t. the ROI classes.

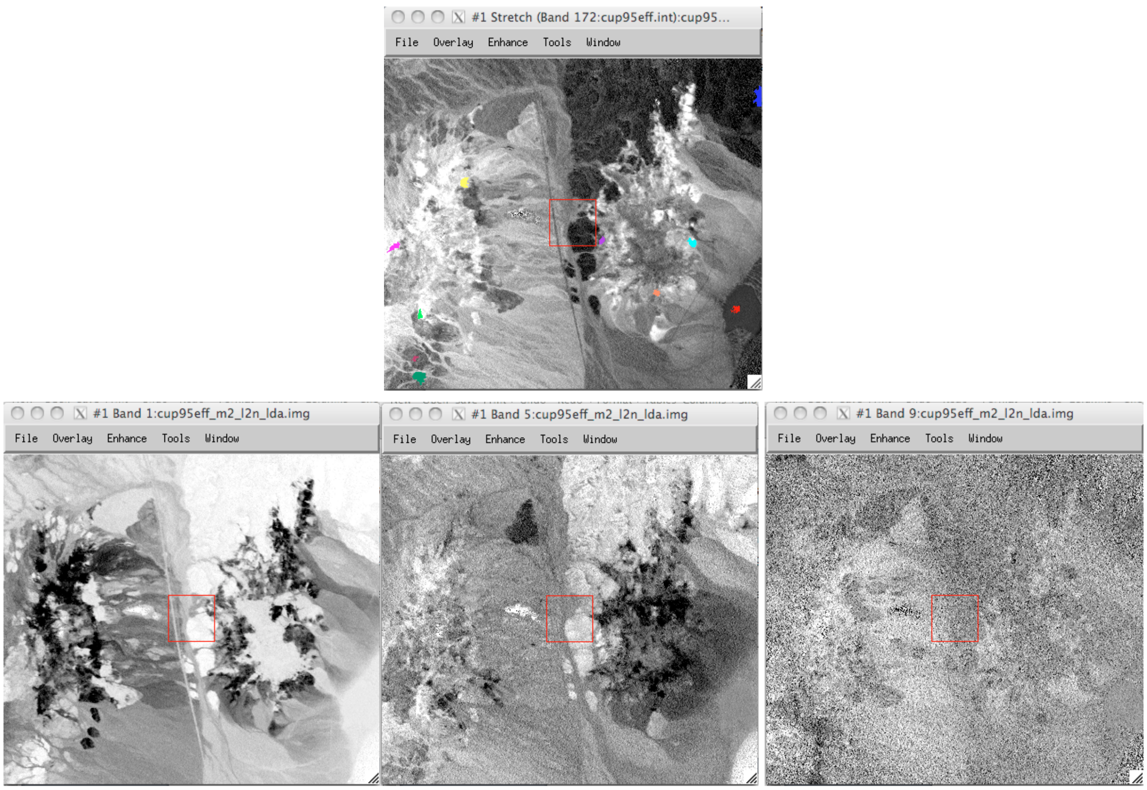

Figure 10: Top: ROIs of 10 SMACC-endmembers. Bottom: Bands 1, 5, and 9 of the transformed image produced by the metric learning function. Bands are ordered according to their informativeness, with the most informative band occurring first.

Contact Information

General HiiHAT questions:	hiihat@jpl.nasa.gov
HiiHAT user mailing list	hiihat-users@jpl.nasa.gov
Software/Bugs/Feature Requests:	hiihat-dev@jpl.nasa.gov
Licensing Requests	hiihat-licensing@jpl.nasa.gov

References

[Bue et al. 2011] B. Bue, D. Thompson, M. Gilmore and R. Castaño "Metric Learning for Hyperspectral Image Segmentation," IEEE WHISPERS 2011, June 2011.

[Thompson et al. 2010] David R. Thompson, Lukas Mandrake, Martha S. Gilmore and Rebecca Castaño. “Superpixel Endmember Detection.” IEEE Transactions on Geoscience and Remote Sensing (to appear).

[Mandrake et al. 2010] Lukas Mandrake, David R. Thompson, Martha S. Gilmore and Rebecca Castaño, Eldar Z. N. Dobrea. “Automated Neutral Region Detection using Superpixels.” IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, July 2010.

[Thompson et al. 2009], David R. Thompson, Martha S. Gilmore and Rebecca Castaño, “Sparse Superpixel Unmixing for Exploratory Analysis of CRISM Hyperspectral Images” IEEE WHISPERS, 2009.

[Gruniger et al. 2004], J. Gruninger, A.J. Ratkowski and M.L. Holke, “The sequential maximum angle convex cone (SMACC) endmember model”. Proceedings of SPIE, 2004.

[Felzenszwalb et al 2004], Pedro F. Felzenszwalb and Daniel P. Huttenlocher, “Efficient Graph-Based Image Segmentation,” International Journal of Computer Vision, Volume 59:2, 2004.

[Winter 1999], M. Winter, “N-FINDR: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. Proceedings of SPIE, 1999.

Sponsor Information / Copyright

This project is sponsored by NASA and the AMMOS MGSS program. Images of Mars from the CRISM instrument, courtesy NASA/JPL/Johns Hopkins University. Spectral data generated using the ENVI/IDL environment.