C

ENVI Extensions in IDL

This appendix provides installation instructions and documentation for all of the ENVI/IDL extensions and their associated routines described in the text.

C.1 Installation

To install the complete extension package:

1. Download and unpack (or clone) the ENVI/IDL package from the GitHub repository, see the links on

   `http://ms-image-analysis.appspot.com/static/homepage/software.html`

   or on

   `http://mcanty.homepage.t-online.de/software.html`

   to any convenient location and point the IDL path to it. Be sure to include its subdirectories.

2. Many of the routines make use of David Fanning's object class

   `CGPROGRESSBAR__DEFINE`

   and associated dependencies, so his `COYOTE` library must also be in the IDL path. The library may be downloaded from

   `http://http://www.idlcoyote.com/`.

3. The IR-MAD scripts require the C-library `PROVMEANS.DLL`. If you are running on a Windows OS, simply point the IDL DLM path to the subdirectory CRC/AUXIL. In addition, if you are using 64-bit IDL and don't have Microsoft Visual C++ installed on your computer, you will need to download and install the freely available Microsoft Visual C++ 2010 Redistributable Package (x64). Unix flavors and Mac OS users will need to create the DLL from the C source code `PROVMEANS.C` provided. See the IDL documentation for `MAKE_DLL`.

4. Under the ENVI Classic GUI, the programs with filenames of the form

   `<program name>_RUN`

   will create their own menu items automatically, so if you wish to call an extension directly from the ENVI Classic menu system, copy (or move) it to the ENVI `SAVE_ADD` directory before invoking ENVI.

5. Under the new ENVI 5 GUI, if you wish the extensions to appear in the Toolbox, copy the programs with filenames of the form

   `<program name>_RUN5`

   to the ENVI `Extensions File Directory`. The programs will appear in the Toolbox organized by book chapter.

   Note: At the time of writing, the ENVI 5.0 API documentation is very obscure in regard to programmatic access to training region data (shape files or ROIs). For this reason, the supervised classification algorithms of Chapters 6 and 7 are only offered for the ENVI Classic interface. Please check the website for eventual updates.

6. (This step is only necessary for the calculation of structure heights with `CALCHEIGHT_RUN` under ENVI Classic and can be skipped if desired.) Copy the file `MJC_CURSOR_MOTION` (in the AUXIL directory) into the `SAVE_ADD` directory. In

   `File/Preferences/User Defined Motion Routine`

   in the ENVI Classic main menu, enter the string: `MJC_CURSOR_MOTION`.

7. Some of the extensions can take advantage of NVIDIA's *Compute Unified Device Architecture* (CUDA), and require the GPULib API available from

   `http:/http://www.txcorp.com/home/gpulib`

   in order to run on the GPU. See the above URL for installation instructions. If CUDA/GPULib is not installed, the scripts will run in the normal way on the CPU.

C.2 Documentation

The ENVI extensions described below can all be run directly from the IDL console, provided that an ENVI interactive session is running. They can also be started from the ENVI Classic Menu or the ENVI 5 Toolbox using the menu items indicated in parentheses.

C.2.1 ENVI extensions for Chapter 4

C.2.1.1 Kernel principal components analysis (Section 4.4.2)

```
KPCA_RUN (Transform/Principal Components/Kernel PCA (CUDA))
KPCA_RUN5 (Toolbox/Extensions/Chapter4/Kernel PCA (CUDA))
```

The user is first queried for an input image filename and a training sample size, maximum 2000. If the latter is 0, then 100 representative training pixel vectors are chosen by the k-means algorithm. Otherwise, a random sample of the desired size is used. Next, the number of kernel principal components to retain (maximum 10) and the kernel parameter `NSCALE` are entered. The Gaussian kernel parameter γ is calculated from the latter as

$$\gamma = \frac{1}{2(\text{NSCALE }\sigma)^2}, \tag{C.1}$$

where $\sigma = \langle \|\boldsymbol{g}(\nu) - \boldsymbol{g}(\nu')\| \rangle$ is the average Euclidean distance between the training observations. The value of γ is printed to the IDL console. Next, the user can choose whether to center the kernel matrix on the training data alone (faster, see Canty and Nieslsen (2012)) or on all of the image pixels. Finally, the output destination (memory or file) is selected.

After centering and diagonalizing the kernel matrix, a plot of the eigenvalues is displayed. Then, if centering on all data was chosen, the projection is carried out in two passes through the image. On the first pass, the matrix column, row and overall sums required for centering are accumulated. These are applied on the second pass as each image pixel is projected. For centering on the training data, only one pass is needed. If CUDA/GPULib are available (this is detected automatically), then both centering and projection are performed on the GPU.

C.2.2 ENVI extensions for Chapter 5

C.2.2.1 Panchromatic sharpening of multispectral images using the discrete or à trous wavelet transform (Sections 5.3.4 and 5.3.5)

```
DWT_RUN (Transform/Image Sharpening/Discrete Wavelet Transform)
DWT_RUN5 (Toolbox/Extensions/Chapter5/DWT Pansharpening)
ATWT_RUN (Transform/Image Sharpening/A Trous Wavelet Transform)
ATWT_RUN5 (Toolbox/Extensions/Chapter5/ATWT Pansharpening)
```

The low- and high-resolution images must be geo-referenced (often they are acquired simultaneously from the same platform) and their spatial resolutions must differ by a factor of 2 or 4. This can be achieved by a prior re-sampling, if necessary. The user is queried for the (spatial/spectral subset of the) image to be sharpened and then for the corresponding panchromatic or high-resolution image. The latter image should overlap the low-resolution image completely.

Then the output destination (memory or file) must be entered. During the calculation, scatterplots are shown of the wavelet coefficients for the low- vs. high-resolution bands.

C.2.2.2 Calculation of the Wang–Bovik index for radiometric fidelity of a pan-sharpened image (Section 5.3.6)

```
QUALITY_INDEX_RUN (Transform/Image Sharpening/Quality Index)
QUALITY_INDEX_RUN5 (Toolbox/Extensions/Chapter5/Quality Index)
```

The user is first prompted for the reference multispectral image to which the sharpened image is to be compared and then for the image whose quality is to be determined. The reference image must completely overlap the pan-sharpened image. The latter is down-sampled to the same resolution as the reference image and then the image bands are matched to the nearest pixel using phase correlation, plots of which are shown during the calculation. The calculated indices are displayed in an IDL text widget.

C.2.2.3 Conversion of SAR imagery to ENVI standard files

```
POLSARINGEST_RUN (Radar/Polarimetric Tools/polSAR Covariance Matrix)
```

Geocoded, multi-look polarimetric SAR imagery suitable for processing with the filtering, classification and change detection algorithms described in the text may be obtained directly from the provider or generated from single-look complex (SLC) data. In the latter case, the open source software packages PolSARpro (European Space agency),

`http://earth.eo.esa.int/polsarpro/`

together with MapReady (Alaska Satellite Facility),

`http://www.asf.alaska.edu/downloads/software_tools`

are a good choice; see Gens et al. (2013). PolSARpro is first used to create multi-look images in covariance matrix format, which can then be exported to MapReady for georeferencing with or without a DEM. Alternative commercial solutions are the Gamma Software (Gamma Remote Sensing)

`http://www.gamma-rs.ch/`

and the SARscape add-on module for ENVI

http://www.exelisvis.com/ProductsServices/ENVI/ENVISARscape.aspx

The ENVI/IDL extension `POLSARINGEST_RUN` combines the outputs from the above preprocessing systems to a single, multi-band ENVI standard file in 32-bit floating point format. For full quad polarimetric data, it generates 9 bands, which are ordered as follows:

$$
\begin{aligned}
&\mathrm{C11} = \langle |s_{hh}|^2 \rangle \\
&\mathrm{C12re} = \langle \sqrt{2} s_{hh} s_{hv}^* \rangle \text{(real part)} \\
&\mathrm{C12im} = \langle \sqrt{2} s_{hh} s_{hv}^* \rangle \text{(imaginary part)} \\
&\mathrm{C13re} = \langle s_{hh} s_{vv}^* \rangle \text{(real part)} \\
&\mathrm{C13im} = \langle s_{hh} s_{vv}^* \rangle \text{(imaginary part)} \\
&\mathrm{C22} = \langle 2|s_{hv}|^2 \rangle \\
&\mathrm{C23re)} = \langle \sqrt{2} s_{hv} s_{vv}^* \rangle \text{(real part)} \\
&\mathrm{C23im)} = \langle \sqrt{2} s_{hv} s_{vv}^* \rangle \text{(imaginary part)} \\
&\mathrm{C33} = \langle |s_{vv}|^2 \rangle .
\end{aligned}
$$

For dual or single polarimetry, the bands corresponding to the missing matrix elements are not present. Thus, a dual polarimetric image will consist of four bands, e.g., C11, C12re, C12im, and C22; a single polarimetric image will consist of just one band, usually C11 or C33. The ENVI files generated by the script can be read by all of the SAR processing routines described below, and also by their Python counterparts described in Appendix D.

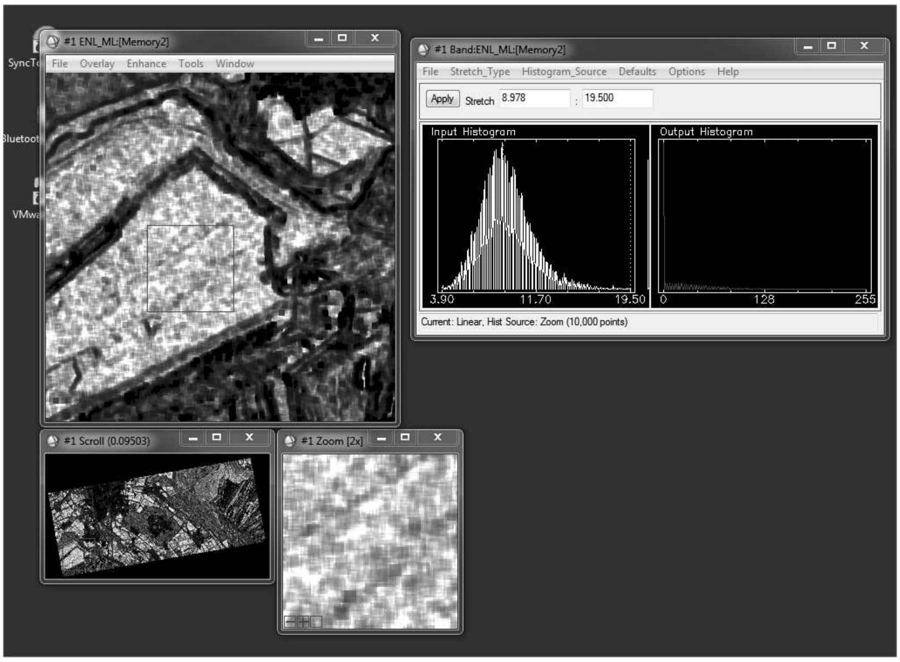
The user is prompted for the directory containing the georeferenced covariance matrix files (consisting of one file for each of the real and imaginary components), and then requested to choose (a spatial subset of) one of them. The other files are then read in automatically with the same spatial subset. Finally, an output filename is requested.

C.2.2.4 Multivariate estimation of equivalent number of looks for polarimetric SAR in covariance matrix format

`ENLML_RUN (Radar/Polarimetric Tools/ENL Estimation)`
`LOOKUP.TXT`

This method is not discussed in the text. It is a multivariate technique based on the maximum likelihood estimator explained in Anfinsen et al. (2009a) and is reported to be superior to the standard univariate method discussed in Section 5.4.2. The polarimetric SAR image should be in the ENVI standard format generated by the routine `POLSARINGEST_RUN` described in Section C.2.2.3 above.

The user is asked to select (a spatial subset of) the ENVI covariance matrix file and the desired window size (default 7 × 7). The local ENL values are estimated in a moving window and stored as a single-band, floating point raster image in ENVI memory. A histogram of the ENL values is plotted. In the ENVI Classic environment, the zoom window can be used to isolate a region in the ENL image with a homogeneous signal background (e.g., flat agricultural field) where the speckle statistics are well developed. ENVI's interactive histogram stretching facility will display a histogram of the ENL



Stretch 8.978 : 19.500

Current: Linear, Hist Source: Zoom (10,000 points)

FIGURE C.1
Illustrating ENL determination in ENVI Classic.

values in the zoom window, the mid-point of which will give an ENL estimate for the full image. An example is shown in Figure C.1.

C.2.2.5 Minimum mean square error filtering of polarimetric SAR imagery (Section 5.4.3.1)

```
MMSE_FILTER_RUN (Radar/Polarimetric Tools/MMSE polSAR filter)
MMSE_FILTER_RUN5 (Toolbox/Extensions/Chapter5/MMSE polSAR filter)
```

The polarimetric SAR image is assumed to have been generated by the `POLSARINGEST_RUN` extension of Section C.2.2.3 above. The user is asked to enter the ENVI filename and the equivalent number of looks. The filtering operation occurs in two steps: first the span image is processed to determine the filter weights, then all components of the covariance matrix are filtered separately. The result may be stored to disk or to ENVI memory.

C.2.2.6 Gamma MAP filtering of polarimetric SAR imagery (Section 5.4.3.2)

```
GAMMA_FILTER_RUN (Radar/Polarimetric Tools/Gamma MAP polSAR filter)
GAMMA_FILTER_RUN (Toolbox/Extensions/Chapter5/Gamma MAP polSAR filter)
```

The polarimetric SAR image is assumed to have been generated by the `POLSARINGEST_RUN` extension of Section C.2.2.3 above. The user is asked to enter the ENVI filename, the equivalent number of looks and, finally, the number of iterations (default 1). Only the diagonal components of the covariance matrix are filtered. The result may be stored to disk or to ENVI memory.

C.2.2.7 Determine heights of vertical structures in high-resolution images (Section 5.5.3)

```
CALCHEIGHT_RUN (Tools/Height Calculator)
```

This routine makes use of rational function models (RFMs) accompanying ortho-ready imagery distributed by the image providers. It is implemented as a small graphical user interface (GUI) using IDL widgets. After displaying a high-resolution image and starting the GUI from the image window's `Tools` menu, an RFM file must be loaded from the GUI menu

```
    File/Load RPC File
```

(extension RPC or RPB). If a digital elevation model (DEM) is available for the scene, this can also be entered with

```
    File/Load DEM File
```

A DEM is not required, however. Click on the bottom of a vertical structure to set the base height and then shift-click on the top of the structure. Press the `CALC` button to display the structure's height, latitude, longitude and base elevation. The number in brackets next to the height is the minimum distance (in pixels) between the top pixel and a (projected) vertical line through the bottom pixel. It should be of the order of 1 or less. If no DEM is supplied, the base elevation is the average value for the whole scene. If a DEM is used, the base elevation is taken from it. The latitude and longitude are then ortho-rectified values.

C.2.2.8 Local solar incidence angle correction for multispectral images over rough terrain (Section 5.5.6)

```
C_CORRECTION_RUN (Topographic/C-Correction)
C_CORRECTION_RUN5 (Toolbox/Extensions/Chapter5/C-Correction)
```

Select the (spectral/spatial subset of the) image to be corrected and (optionally) a mask. Then, in the edit box, enter the solar elevation and azimuth in degrees and a size for the kernel used for slope/aspect determination (default 9×9). Then select the corresponding DEM file and the output destination (file or memory).

C.2.2.9 Tie-point extraction for image–image registration (Section 5.6.2)

```
CONTOUR_MATCH_RUN (Map/Registration/Contour Matching)
CONTOUR_MATCH_RUN5 (Toolbox/Extensions/Chapter5/Contour Matching)
```

This extension is primarily intended for fine adjustment, assuming that the images are already geo-referenced and, if necessary, re-sampled to the same ground sample distance. From the ensuing dialog windows, choose the base and warp image bands. Then enter the σ parameter for the LoG filter (default value is 2.5 pixels). Finally, give a filename (extension `.PTS`) for the tie-points (ground control points). If tie-points are found, they can be read from this file into ENVI's `Ground Control Points Selection` dialog. It may be necessary to edit them further to eliminate outliers.

C.2.3 ENVI extensions for Chapter 6

C.2.3.1 Bayes maximum likelihood classifier (Section 6.3.2)

```
MAXLIKE_RUN (Classification/Supervised/MaxLike (test output))
```

This routine is a wrapper for ENVI's maximum likelihood classifier, which reserves test data from the training regions of interest (ROIs) in the ratio 2:1 for training:test. The user is prompted for an input file, the training ROIs (which must be present beforehand) and output destinations for the classified image, the rule image (optional) and the test results. The test result file can be processed with the procedures `CT_RUN` and and `McNEMAR_RUN` discussed in Section C.2.4.2 below. The routine `RULE_CONVERT`, callable only from the ENVI command prompt, may be used to convert the ENVI rule image to probabilities; see Section 6.3.1.

C.2.3.2 Gaussian kernel classification (Section 6.4)

```
KERNEL_RUN (Classification/Supervised/Gaussian Kernel (CUDA))
```

This extension performs nonparametric classification of a multispectral image using the Gaussian kernel algorithm. The program will make use of CUDA, if it is installed, to calculate the Gauss kernel matrices. The user is prompted for an input file, the training ROIs (which must be present beforehand) and output destinations for the classified image, the probability image (optional), and the test results (optional). During calculation, a plot window shows the minimization of the misclassification rate with respect to the parameter σ as shown in Figure 6.5. The program uses spectral tiling and may be used for large images. The test result file can be processed with the procedures `CT_RUN` and `McNEMAR_RUN` discussed in Section C.2.4.2 below.

C.2.3.3 Neural network classifiers (Section 6.5 and Appendix B)

`FFN_RUN (Classification/Supervised/Neural Net (Kalman/Conj Gradient))`

The extension `FFN_RUN` classifies a multispectral image with a two-layer, feed-forward neural network trained with a combination of the Kalman filter algorithm and the scaled conjugate gradient algorithm. The user is prompted for an input file, the training ROIs (which must be present beforehand), output destinations for the classified image, the probability image (optional) and the test results (optional), the number of hidden neurons and, finally, whether or not to use validation. During the calculation, the cross-entropy cost function is displayed. Training can be interrupted at any time, whereupon processing continues to the next phase. After training, the Hessian matrix may be calculated, in which case its eigenvalues are written to a text window. The program uses spectral tiling and may be used for large images. The test result file can be processed with the procedures `CT_RUN` and and `McNEMAR_RUN` discussed in Section C.2.4.2 below.

C.2.3.4 Support vector machine classifier (Section 6.6.6)

`SVM_RUN (Classification/Supervised/SVM (test output))`

This routine is a wrapper for ENVI's support vector machine classifier, which reserves test data from the training regions of interest (ROIs) in the ratio 2:1 for training:test. The user is prompted for an input file and the training ROIs (which must be present beforehand) and output destination for the test results. Classification and rule images are written to memory only. SVM-specific parameters (kernel type, kernel parameters, pyramid depth) must be set programmatically. The test result file can be processed with the procedures `CT_RUN` and `McNEMAR_RUN` discussed in Section C.2.4.2 below.

C.2.4 ENVI extensions for Chapter 7

C.2.4.1 Probabilistic label relaxation postprocessing (Section 7.1.2)

```
PLR_RUN (Classification/Post Classification/PLR)
PLR_RECLASS_RUN (Classification/Post Classification/PLR Reclassify)
PLR_RUN5 (Toolbox/Extensions/Chapter7/PLR)
PLR_RECLASS_RUN5 (Toolbox/Extensions/Chapter7/PLR Reclassify)
```

The `PLR_RUN` or `PLR_RUN5` extension takes as input a class probability vector image (rule image) generated by any of the supervised classification extensions described in this appendix, as well as from the clustering routine `EM_RUN` described in Section C.2.5.4 below, and generates a modified class probability vector image. At the prompt, choose a class membership probabilities image and the number of iterations (default = 3). Then choose a destination either in memory or as a named file.

The generated rule image can then be processed with `PLR_RECLASS_RUN` or `PLR_RECLASS_RUN5` to generate an improved classification image with better spatial coherence. At the prompt, choose a previous classification image if available (to synchronize the class colors) and then a class membership probabilities image. At the next prompt for a dummy unclassified class reply "Yes." Then choose a destination either in memory or as a named file.

C.2.4.2 Contingency tables and McNemar test (Section 7.2)

```
CT_RUN (Classification/Post Classification/Contingency Table)
MCNEMAR_RUN (Classification/Post Classification/McNemar Test)
CT_RUN5 (Toolbox/Extensions/Chapter7/Contingency Table)
MCNEMAR_RUN5 (Toolbox/Extensions/Chapter7/McNemar Test)
```

The procedures `CT_RUN(5)` and `McNEMAR_RUN(5)` are used to evaluate and compare test results generated by the supervised classifiers `MAXLIKE_RUN`, `KERNEL_RUN`, `FFN_RUN`, `FFN3AB_RUN` (see below) and `SVM_RUN`. They request input files with extension `TST` and write their outputs (contingency tables, accuracies, test statistics etc.) to the IDL console.

C.2.4.3 AdaBoost.M1 algorithm with three-layer neural network classifiers (Section 7.3)

```
FFN3AB_RUN (Classification/Supervised/Neural Net (AdaBoost))
```

The user is asked to choose an input file, the training ROIs (which must be present beforehand), output destinations for the classified image, the probability image (optional) and the test results (required). Then the user is prompted for the number of training epochs for each network in the sequence (default 5) and the numbers of neurons in each of the two hidden layers (default 10). During the calculation, the cross-entropy cost function for the current network is displayed as well as the overall training and generalization errors of the sequence; see Figure 7.3. Training can be interrupted at any time, whereupon the boosted sequence is terminated and the image is classified. The program uses spectral tiling and may be applied to large images. The test result file can be processed with the procedures `CT_RUN(5)` and `McNEMAR_RUN(5)` discussed above.

C.2.4.4 Maximum likelihood classification of polarimetric SAR imagery (Section 7.4)

```
MAXLIKESAR_RUN (Classification/Supervised/MaxLike polSAR)
```

The quad or dual polarimetric SAR image should have the been generated by the `POLSARINGEST_RUN` extension of Section C.2.2.3. The user is asked to enter the ENVI filename, then prompted for the training ROIs (which must be present beforehand), and finally for a filename for the test statistics. The classification result is written to ENVI memory only, but can be saved to disk manually. The test statistics file can be processed with the procedure `CT_RUN` or `CT_RUN5` discussed in Section C.2.4.2 above.

C.2.4.5 Kernel RX anomaly detection (Section 7.5.5)

```
KRX_RUN (Spectral/Kernel RX (CUDA))
KRX_RUN5 (Toolbox/Extensions/Chapter7/Kernel RX (CUDA))
```

Choose (a spatial/spectral subset of) a multi- or hyperspectral image in the first dialog. Then select the number of training samples for the kernel matrix (select 0 to determine them with the K-means algorithm). Next, enter the `NSCALE` parameter for the Gaussian kernel (the Gaussian kernel parameter γ is calculated as in Equation (C.1)), and finally the output destination for the anomaly image. If CUDA/GPULib is available, the anomaly values are calculated on the GPU.

C.2.5 ENVI extensions for Chapter 8

C.2.5.1 Kernel K-means clustering of multispectral imagery (Section 8.2.2)

```
KKMEANS_RUN (Classification/Unsupervised/Kernel K-Means (CUDA))
KKMEANS_RUN5 (Toolbox/Extensions/Chapter8/Kernel K-Means (CUDA))
```

Choose (a spatial/spectral subset of) a multispectral image in the first dialog. Then enter the number of random training samples and number of clusters in the corresponding prompts. The next prompt is for the kernel parameter `NSCALE` (the Gaussian kernel parameter γ is calculated as in Equation (C.1)). Finally, decide whether or not to save the clustered training data as ROIs and choose the output destination (memory or file). If CUDA/GPULib is available, then both training and generalization (classification) are performed on the GPU, resulting in a considerable speedup. This extension does not make use of the ENVI tiling facility and is not intended to be used with large images.

C.2.5.2 Agglomerative hierarchical clustering of multispectral imagery (Section 8.2.4)

```
HCL_RUN (Classification/Unsupervised/Agglomerative Hierarchical)
HCL_RUN5 (Toolbox/Extensions/Chapter8/Agglomerative Clustering)
```

Clustering is performed on a small random sample of image pixels (≤ 2000), and generalized to the entire image by using ENVI's built-in maximum likelihood supervised classification algorithm. Choose (a spatial/spectral subset of) a multispectral image in the first dialog. Then enter the number of random samples and number of clusters in the corresponding prompts. Finally, indicate whether the classified image is to be saved to disk or memory. During computation, the current number of clusters is displayed in the progress bar. The result can be used to cluster a larger spatial subset if so desired (this is prompted for). This extension does not make use of the ENVI tiling facility and is not intended to be used with large images.

C.2.5.3 Fuzzy K-means clustering of multispectral imagery (Section 8.2.5)

```
FKM_RUN (Classification/Unsupervised/Fuzzy K-Means)
FKM_RUN5 (Toolbox/Extensions/Chapter8/Fuzzy K-Means Clustering)
```

Clustering is performed on a random sample of image pixels ($\leq 10^5$), and generalized to the entire image by using a modification of the IDL distance classifier `CLUSTER_FKM.PRO` (in the AUXIL directory). Choose (a spatial/spectral subset of) a multispectral image at the prompt. Then enter the desired number of clusters and whether or not they should be saved as ROIs at the corresponding prompts. Finally, indicate whether the class image is to be saved to disk or memory. This extension does not make use of the ENVI tiling facility and is not intended to be used with large images.

C.2.5.4 Gaussian mixture clustering using the expectation maximization algorithm (Section 8.3)

```
EM_RUN (Classification/Unsupervised/Gaussian Mixture (CUDA))
EM_RUN5 (Toolbox/Extensions/Chapter8/Gaussian Mixture Clustering (CUDA))
```

Clustering occurs (optionally) at different scales, and both simulated annealing and spatial memberships can be included if desired. The program will also optionally generate class membership probability images, which can be postprocessed with probabilistic label relaxation (see Section C.2.4.1). Choose (a spatial/spectral subset of) a multispectral image at the prompt. Then enter the desired number of clusters and compressions (pyramid depth) at the corresponding prompts. In the `Clustering parameters` dialog, enter the initial annealing temperature (zero for no annealing) and spatial membership parameter `Beta` (zero for no spatial memberships). At the next prompt, indicate whether or not to save the clusters as ROIs. Finally, say whether the class image is to be saved to disk or memory and similarly for the membership probability image (optional). This extension does not make use of the ENVI tiling facility and is not intended to be used with large images.

C.2.5.5 Image cluster visualization using the Kohonen self-organizing map (Section 8.6)

```
SOM_RUN (Classification/Unsupervised/Self Organizing Map)
SOM_RUN5 (Toolbox/Extensions/Chapter8/Self Organizing Map)
```

At the prompts, enter the (spectral/spatial subset of the) image to be visualized, the dimension of the neural network cube (default $6 \times 6 \times 6$), and the output destination for the map. The network is trained with a random sample of 10^4 pixel vectors. A three-dimensional plot of the neuron cube projected onto the space of the first three image bands is generated; see Figure 8.7. The routine does not use the ENVI tiling facility for the final classification and is not intended to be used with large images.

C.2.5.6 Segmentation of a class image by finding connected components for each separate class (Section 8.7.1)

```
SEGMENT_CLASS_RUN (Classification/Unsupervised/Cluster Segments)
```

This routine is an alternative to the ENVI `.../Segmentation Image` command and, unlike that command, honors class labels. At the prompts, enter the (spatial subset of the) classification image to be segmented and select the classes to include in segmentation. Then choose the minimum segment size to accept, the connectivity (four or eight) for blobbing and, finally, the output destination (memory or file).

C.2.5.7 Mean shift segmentation of multispectral images (Section 8.7.3)

```
MEAN_SHIFT_RUN (Classification/Unsupervised/Mean Shift Segmentation)
MEAN_SHIFT_RUN5 (Toolbox/Extensions/Chapter8/Mean Shift Segmentation)
```

The user is prompted for the (spectral/spatial subset of the) multispectral (or gray-scale) image to be segmented, and then, in an edit widget, for the spatial and spectral bandwidths (defaults 15) and minimum segment size (default 30 pixels). Finally, the user must choose a destination (memory or file) for the output. Three images are generated:

1. The segmented image (ENVI standard)
2. The segment boundaries (ENVI classification file)
3. The segment labels (ENVI standard)

The boundaries can be overlaid onto the segmented image (or segment labels) with the `Overlay/Classification` command in the ENVI Classic image menu. The routine associates pixels along the current mean shift path with the final mode reached on that path, leading to an unavoidable fragmentation of some of the segments. To counter this effect to some extent, the segmented image is postprocessed with a 3×3 median filter.

C.2.6 ENVI extensions for Chapter 9

C.2.6.1 Iteratively re-weighted multivariate alteration detection (Section 9.4)

```
MAD_RUN (Basic Tools/Change Detection/iMAD/Compute new stats...)
MAD_RUN5 (Toolbox/Extensions/Chapter9/Compute new stats AND run iMAD)
MAD_VIEW_RUN (Basic Tools/Change Detection/MAD View)
```

As with other linear spectral transformations in the ENVI environment such as PCA and MNF, there are two ways to run iMAD:

1. by computing image statistics from the current data, i.e., the bitemporal image itself, or
2. by reading an existing statistics file and applying it to the current data.

Method 1

The first of these is the more common. Begin by choosing (spatial/spectral subsets of) two multispectral images at the prompts. When entering the first image, a mask band can be optionally specified. Then enter the maximum number of iterations (default 50), a penalization regularization parameter (default 0.0) and the output destinations for the MAD and canonical variates (CVs). Press `cancel` for whichever output is not desired. If the CVs are to be output to a file, the program will append _1 and _2 to the filenames in order to distinguish the two CV files which are created. If the two images are not of the same spectral/spatial dimension, the program will abort. If an image is not in band interleaved by pixel (BIP) or band interleaved by line (BIL) format, you will be prompted to allow it to be converted to BIP in place (the image must reside on disk in this case). Finally, you are prompted for a statistics filename (ASCII file with extension `MAD`) where you can (optionally) save the transformation coefficients calculated with the current data. The program also appends an additional band to the MAD variates image consisting of the chi-square value (sum of squares of standardized MAD values) of each pixel.

There are two primary reasons for masking the images:

1. "Black edge" pixels. Often, when processing full scenes, the image margins contain no data. The MAD algorithm may then misinterpret these pixels as the no-change background and converge to them. Since the edge pixels are constants (usually zeroes), the weighted covariance matrices will quickly become degenerate and the program will abort, usually with the message `CHOLDC FAILED`.
2. Large water bodies. Generally, water bodies provide a good no-change background against which to measure change. However, if they are large and if illumination effects (e.g., due to waves, solar glare) lead to a uniform difference in reflectance between the two acquisitions, then

they can constitute a false no-change background to which the MAD algorithm may converge.

Both effects can be countered by appropriate masking. Cloud cover, on the other hand, is usually never a problem, since it corresponds to genuine change.

Method 2

The second method is provided primarily for situations in which the MAD algorithm fails to converge because the amount of real change between the acquisitions is too large: a useful no-change background is not found. If this is the case, it may be possible, by experimentation, to find a spatial subset of the data for which convergence is possible. The generated statistics file (see above) may then be used to generalize to the entire scene. After responding to the prompt for an existing stats file, enter the (spatial/spectral) subsets of the two images (the spectral subsets must, of course, concur with those used to generate the stats file) and the destinations for the MAD and canonical variates. The MAD transformation is then performed immediately.

These extensions use ENVI spectral tiling and can be run on large datasets, e.g., Landsat TM full scenes.

Viewing

`MAD_VIEW_RUN` is a simple ENVI/IDL GUI for viewing MAD or MAD/MNF variates. It can be used to set marginal thresholds on any three MAD variates simultaneously and to stretch the data over different dynamic ranges. It is provided with a rudimentary help file in PDF format.

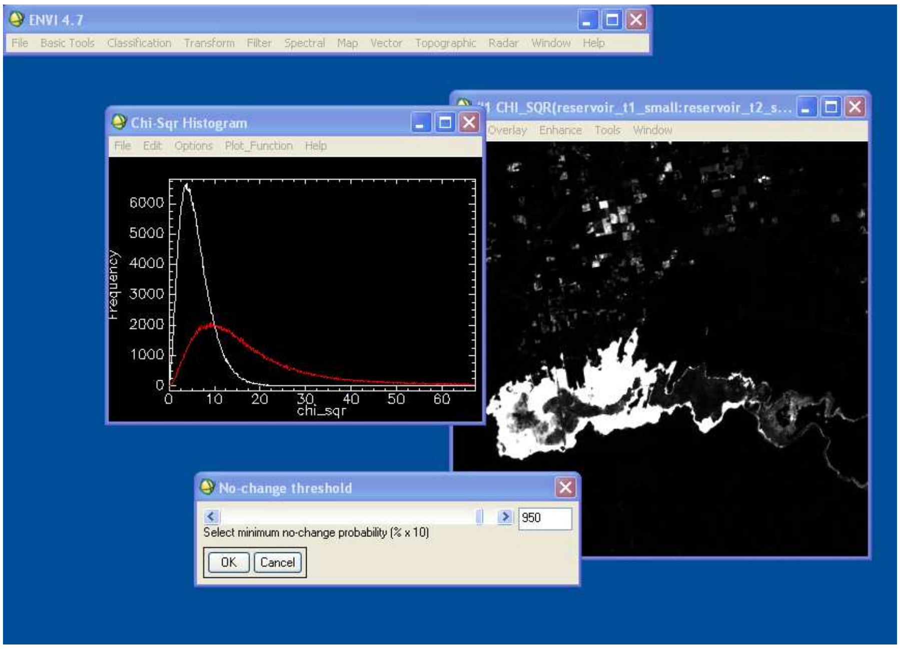
C.2.6.2 Polarimetric SAR Change Detection

```
WISHARTCHANGE_RUN (Radar/Polarimetric Tools/SAR Change Detection)
WISHARTCHANGE_RUN5 (Toolbox/Extensions/Chapter9/SAR Change Detection)
```

The two polarimetric SAR images should have been generated by the `POLSARINGEST_RUN` extension of Section C.2.2.3, and not necessarily have been coregistered. The user is asked to enter the ENVI filename and the equivalent number of looks for both images. The images must have the same polarimetry, which can be any of the following:

- full polarimetry
- dual polarimetry
- single polarization

Then, a desired significance level for change identification (default 0.01) and output destination (ENVI memory or disk) must be chosen. Two image files are created:


ENVI 4.7
File Basic Tools Classification Transform Filter Spectral Map Vector Topographic Radar Window Help
Chi-Sqr Histogram
File Edit Options Plot_Function Help
Frequency
6000
5000
4000
3000
2000
1000
0
0 10 20 30 40 50 60
chi_sqr
#1 CHI_SQR(reservoir_t1_small:reservoir_t2_s...
Overlay Enhance Tools Window
No-change threshold
OK
Cancel

Select minimum no-change probability (% x 10) 950

FIGURE C.2
Radiometric normalization. **(See color insert.)**

- a 4-band image consisting of the span of the first image chosen, the test statistic $-2\rho \ln Q$, the associated change probability $\Pr(-2\rho \ln Q) \leq z$ and the thresholded change probability at the desired significance;

- an ENVI classification image for change and no-change at the chosen significance level.

C.2.6.3 Scatterplot matching (Section 9.8.1)

```
BSLFCP_RUN (Basic Tools/Change Detection/Scatterplot Normalization)
BSLFCP_RUN5 (Toolbox/Extensions/Chapter9/Scatterplot Normalization)
```

This extension exploits the characteristic shape of NIR vs. red scatterplots in multispectral images for radiometric normalization. The method only applies for the two bands in question, e.g., bands 3 and 4 for Landsat TM imagery or bands 2 and 3 for ASTER VNIR data. The user is prompted for the red and near-infrared bands of the reference and target image and the output destination (memory or file). Scatterplots of the reference image and of the target image before and after normalization are displayed and the result written as a two-band ENVI standard image.

C.2.6.4 IR-MAD relative radiometric normalization (Section 9.8.2)

`RADCAL_RUN (Basic Tools/Change Detection/Radiometric Normalization)`
`RADCAL_RUN5 (Toolbox/Extensions/Chapter9/Radiometric Normalization)`

This extension takes advantage of the linear and affine invariance of the MAD transformation to perform a relative radiometric normalization of the images involved in the transformation.

At the prompts, choose the same spatial subset of the two images as were used in Method 1 in Section C.2.6.1 above to generate the IR-MAD image. The spectral subsets must be equal, but need not be the same as for the IR-MAD image. Then enter the chi-square band which was appended to the MAD image. You will then be prompted to set a no-change probability threshold for the pixels to be used in the normalization (default 95%); see Figure C.2. The plot on the left shows, in white, the theoretical chi-square distribution for no-change data and, in red, the actual distribution as determined for all of the pixels in the image. The threshold probability in the slider refers to the theoretical distribution. If the true distribution is severely shifted away from the theoretical one due to a predominance of change, it may be necessary to choose a lower threshold in order to have sufficiently many no-change pixels for a good normalization.

The next prompt is for a destination (memory or disk) for the normalized image, after which the normalization takes place. Orthogonal regression lines are plotted for each multispectral band and a table showing associated statistical parameters and some hypothesis test results is presented.

A final prompt asks if the current regression parameters are to be applied in order to normalize another file (e.g., a full scene). If so, that file must be chosen with the same spectral subset as for the reference and target images and a destination filename must be supplied.