HYperspectral Data Retrieval and Analysis (HYDRA): Synthesis tools package

Computational Intelligence Group
Basque Country University (UPV/EHU)
http://www.ehu.es/computationalintelligence

November 5, 2010

Contents

Chapter 1

Introduction

This document introduces the Synthesis tools package of the HYDRA project. HYDRA is an on-work development by the Computational Intelligence group (http://www.ehu.es/ccwintco) of the Basque Country University (http://www.ehu.es).

HYDRA is a set of tools focused in the development of computational methods for the analysis of hyperspectral imagery and the information retrieval from large hyperspectral databases. The synthesis tools package is part of the HYDRA project and it groups a set of methods to make in-lab synthetic hyperspectral images.

HYDRA is an open source project under the GNU GPL3 license. A copy of the license should be included with the utilities.

The Synthesis tools package has been actually developed with MATLAB 7.4 software (http://www.mathworks.com/) and a licensed copy is needed to use it.

Chapter 2

Synthesis tools package description

2.1 Installation

You can download the Synthesis tools package from the Computational Intelligence group webpage: `http://www.ehu.es/ccwintco/XXX`.

To make it available from MATLAB:

1. Unzip the downloaded file "synthesis.zip". This will create a folder named "synthesis".
2. Run the MATLAB software (Figure 2.1).
3. In the menu press "File->Set Path...". This will open the "Set Path" window (Figure 2.2).
4. In the "Set Path" window press "Add with subfolders" and select the "Synthesis" folder (Figure 2.3).
5. Press the "Save" buttom and close the "Set Path" window (Figure 2.4).

The Synthesis package has the following structure:

- Synthesis
  - Endmembers
    - USGS Spectral library (inside "USGS Spectral library" folder there are the original spectral library from the USGS lab).
    - endmembers.mat (a selection of spectras for direct use with MATLAB).
    - endmembersreflectance.m (a Matlab file to extract the endmembers reflectance to a ".mat" file).

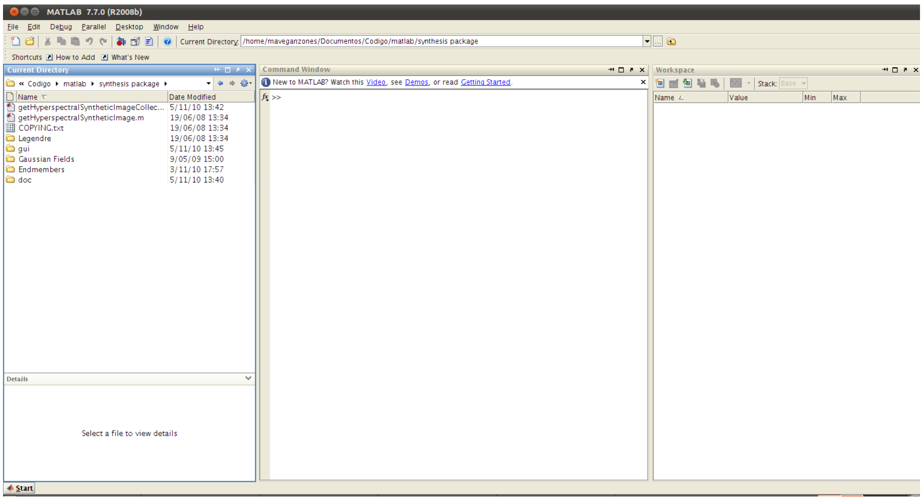

Figure 2.1: The MATLAB software.

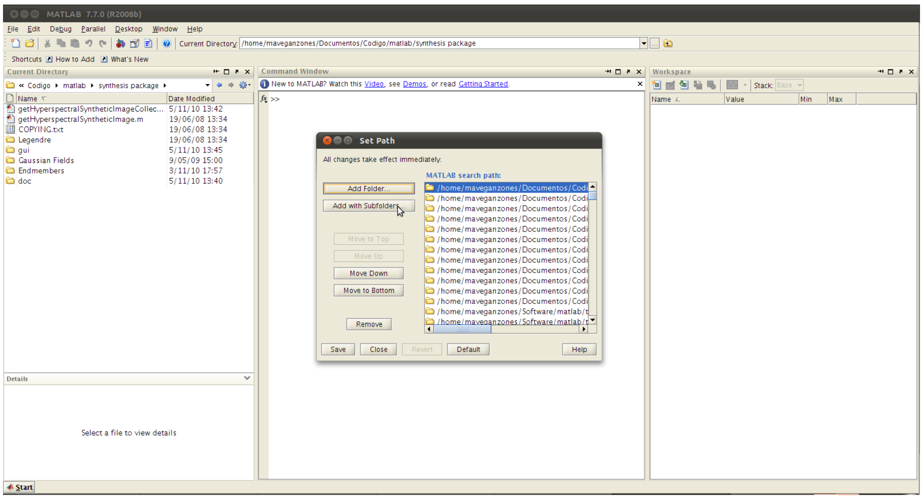

Figure 2.2: The “Set Path” Window.

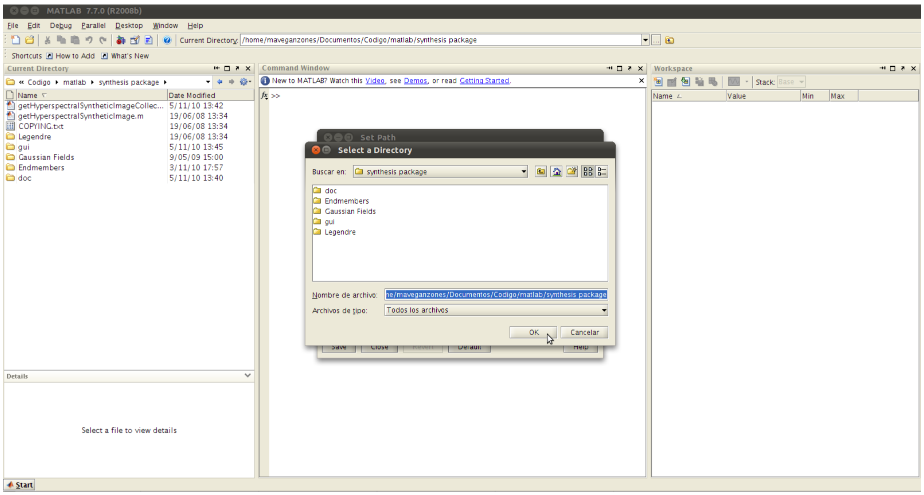

Figure 2.3: Selecting the “synthesis” package.

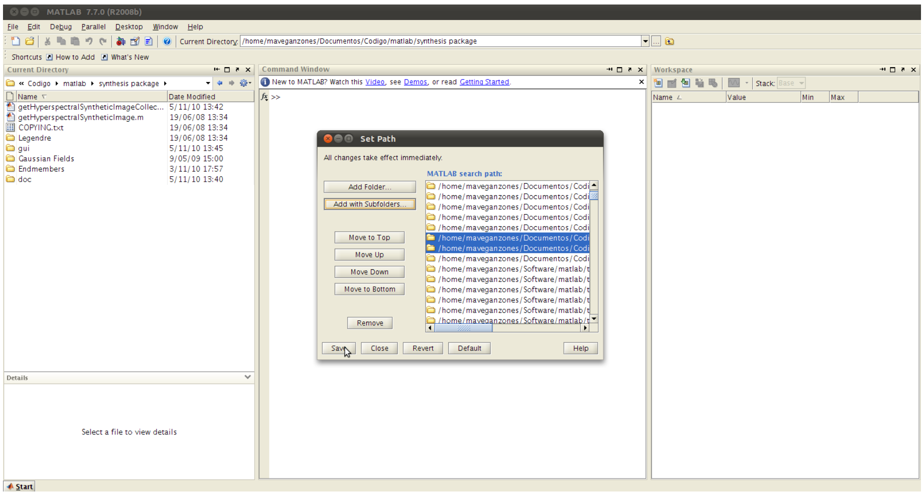

Figure 2.4: The “synthesis” package has been added to the Matlab path.

    * endmembersreflectance.mat (the reflectance values of the selection of endmembers for direct use with MATLAB)

  - Gaussian Fields (inside “Gaussian Fields” folder there are the source files for the Gaussian Fields based synthetic method).
  - Legendre (inside “Legendre” folder there are the source files for the Legendre based synthetic method).
  - COPYING.txt (the GNU GPL3 license).
  - manual.pdf (this document).

2.2 The USGS spectral library

Researchers at the USGS Spectroscopy Lab have measured the spectral reflectance of hundreds of materials in the lab, and have compiled a spectral library. The libraries are used as references for material identification in remote sensing images. The website of the USGS Spectroscopy lab is `http://speclab.cr.usgs.gov/`.

The UV to mid-infrared 0.2-150 micron library (splib06) contains over 1300 spectra including mid-infrared data as well as spectra from splib05a and additional visible and near-infrared spectra. The library includes many more minerals, organic and volatile compounds, vegetation, and man-made materials than our previous libraries.

The full USGS Spectral library is included with the Synthesis package, but the most recent library is always available throught the USGS Spectroscopy lab: `http://speclab.cr.usgs.gov/spectral-lib.html`.

A selection of 20 minerals endmembers is included with the Synthesis package as a “endmembers_samples.mat” file for direct use. This file, localized in the “endmembers” folder, loads 20 matrix each correponding to a mineral endmember (Figure 2.5). Each matrix has 2151 rows and 3 columns, being each row a measure for each wavelenght and the columns the following respectively measures: the wavelength value, the reflectance and the variance observed in the measuring proccess. reflectance is the relevant value to work in the synthetic proccess (the second column). Note that $-1.23\exp^{34}$ indicates a deleted number, and that standard deviations of 0.000000 means not measured.

Together with the sample endmembers “mat” file, a “getEndmembers.m” function is provided to extract the information on the USGS folders to a matlab variable. This function use is explained in chapter 4.

If more information is needed, please see the USGS data files or visit the USGS official web site.

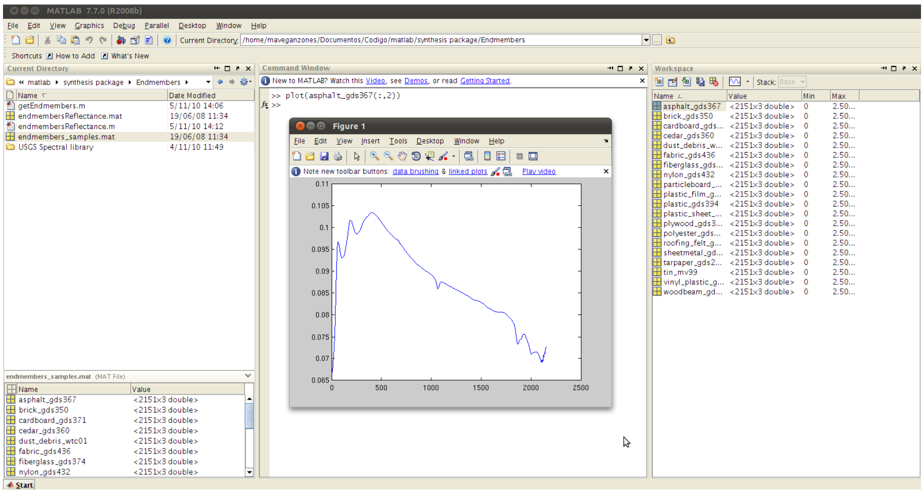

Figure 2.5: Endmembers from the “endmembers_sample.mat” file loaded in the MATLAB workspace. A plot of the asphalt_gds367 reflectance is showed.

Chapter 3

Use examples

3.1 Loading the endmembers

- Loading the included selection of endmembers from the USGS Spectral library:
  1. Enter the “Endmembers” folder into the “Synthesis” package from the Matlab path (Figure 3.1).
  2. Into the folder panel on the left “double click” over the “endmembers_samples.mat” file. This will load the set of spectras to the workplace. Alternativelly, you can load them from the console or a “.m” file with the command: “load endmembers_samples.mat”.
  3. Select the “workspace” tab in the left panel to see the loaded set of endmembers (Figure 3.2).

- To group the reflectance values of the set of endmembers, or a subset, the “endmembersReflectance.m” file can be used or modified (Figure 3.3). This file is inside the “Endmembers” folder. To use it just type the command:

  ```
  endmembersReflectance
  ```

  This command will create an “endmembersReflectance.mat” file that could be loaded to get a matrix with the endmembers reflectance (Figure 3.4). An “endmembersReflectance.mat” file with the full selected endmembers reflectance is also included for direct use.

- Sometimes the use of the full spectra range (the 2152 bands) is not necesary or it could be computationally overcosting. To select a uniform range of bands all along the spectra use the command

  ```
  endmembersReduced = endmembers(:,1:n:2151)
  ```

  where n is the selecting ratio. In a plot of the same material spectra with the full range and the reduced one, no significant differences are visually apreciated (Figure 3.1).

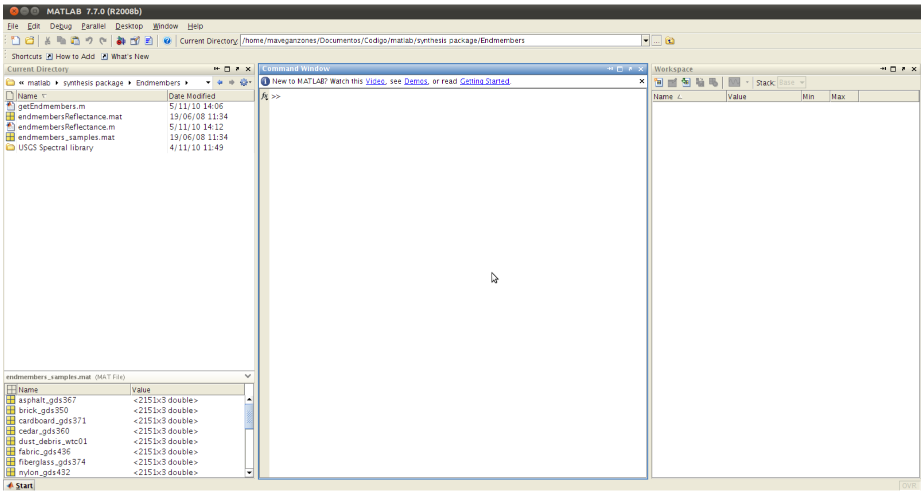

Figure 3.1: Into the “Endmembers” folder.

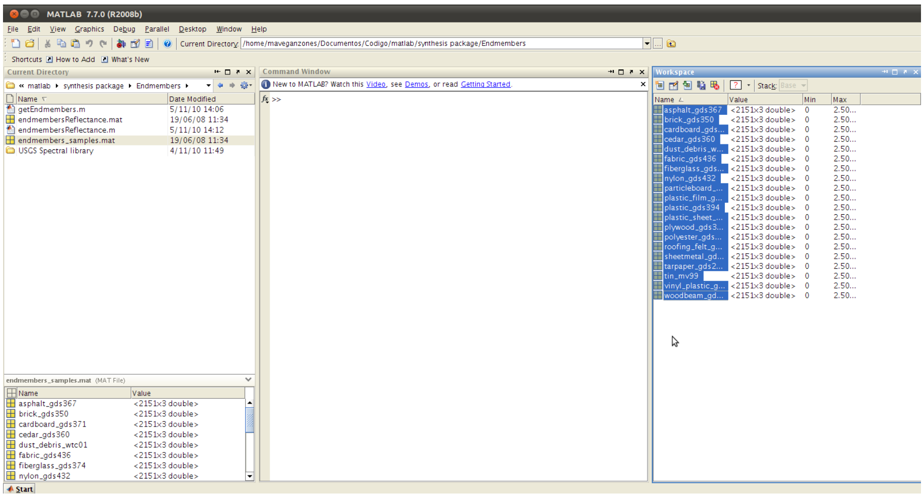

Figure 3.2: The endmembers showed in the workspace tab.

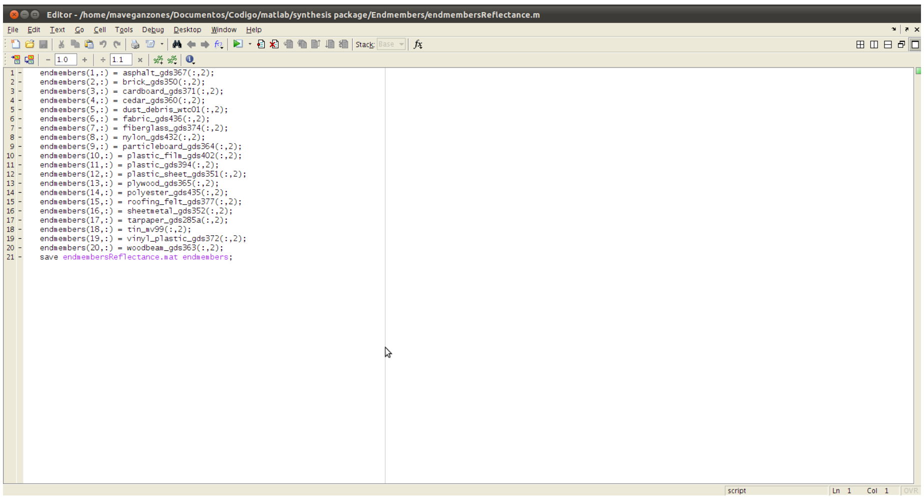

Figure 3.3: The endmembersReflectance.m file.

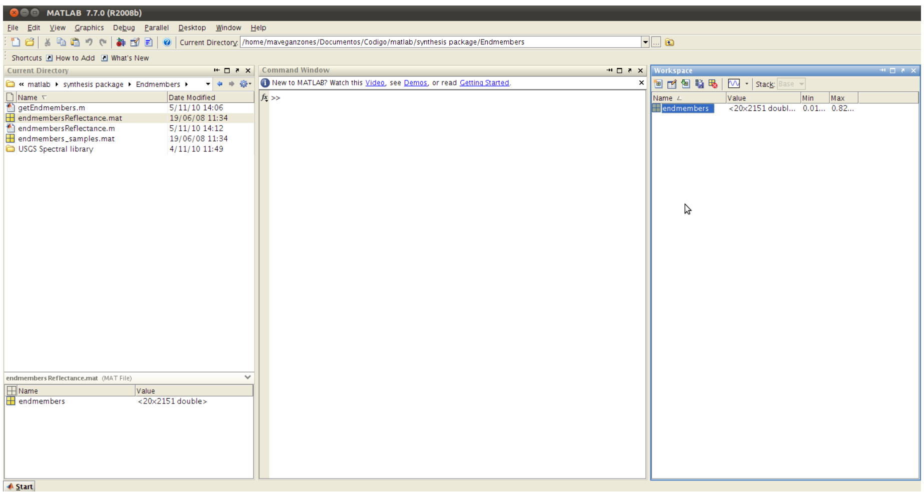

Figure 3.4: The endmembers reflectance matrix loaded by endmembersReflectance.mat file.

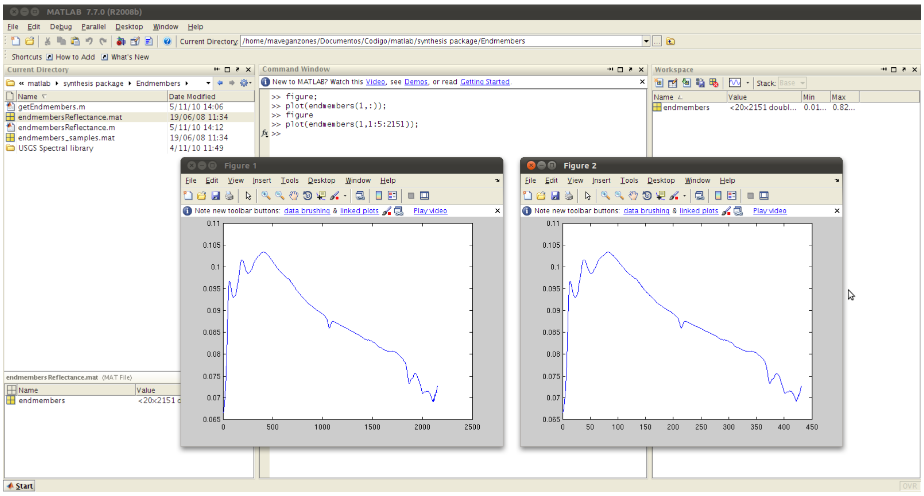

Figure 3.5: Plot of a material spectra with the full range (left) and the reduced one (right).

3.2 Making synthetic abundancies

3.2.1 Legendre method

- To build a synthetic abundancies sample with the Legendre method just type the command:

```
[abundanciesSample] = getAbundanciesSampleLegendre();
```

  that will create a sample of abundancies images with the default parameters (Figure 3.6). The "abundanceSample" will be a matrix of $256 \times 256 \times 5$ elements, being the first and second the spatial dimensions and the last the number of abundancies images.

  The parameters used by the Legendre method are:

  - ne: number of abundancies images in the sample (default: 5).
  - npoints: spatial dimensions of the abundancies images, the images are squared so only one parameter is needed (default: 256).
  - minOrder,maxOrder: minimum and maximum orders of the Legendre polynomies (defaults: 1,10).
  - maxCoef: weighting coeficient to bias the Legendre polynomies (default: 100).

  An use example could be:

```
[abundanciesSample] =
getAbundanciesSampleLegendre(3,64,1,5,50);
```

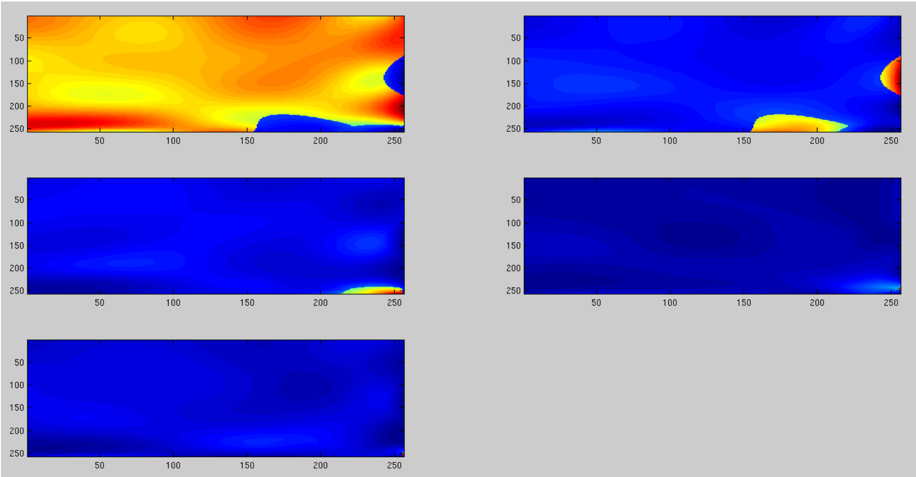

Figure 3.6: Sample of abundancies images made by Legendre method (default parameters).

that will create a sample of 3 abundancies images with 64 pixels-height and width, and with minOrder, maxOrder and maxCoef equals to 1, 10 and 50 respectively (Figure 3.7). In this case the "abundanceSample" will be a matrix of $64 \times 64 \times 3$ elements, being the first and second the spatial dimensions and the last the number of abundancies images.

- To build a collection of synthetic abundancies samples with the Legendre method just use the command:

```
[abundanciesCollection] = getAbundanciesCollectionLegendre(n);
```

that will create a collection of n Legendre abundancies samples with the default parameters. The "abundanciesCollection" will be a matrix of $n \times 256 \times 256 \times 5$ elements, being the first the sample index, the second and third the spatial dimensions of each sample and the last the number of abundancies images.

To control the parameters of the Legendre method, just put the parameters explained above after the n parameter. An example:

```
[abundanciesCollection] =
getAbundanciesCollectionLegendre(20,3,64,1,5,50);
```

will create a collection of twenty Legendre samples with 3 abundancies images each. In this case the "abundanciesCollection" will be a matrix of $20 \times 64 \times 64 \times 3$ elements, being the first the sample index, the second and third the spatial dimensions of each sample and the last the number of abundancies images.

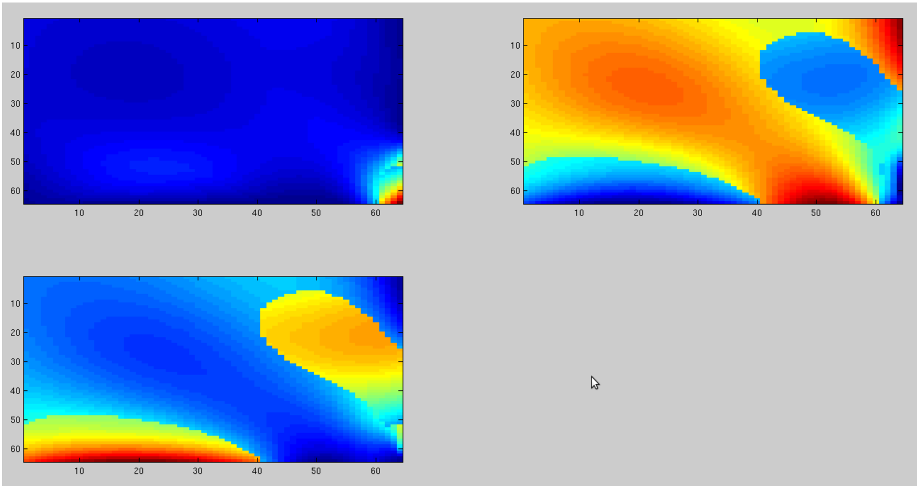

Figure 3.7: Sample of abundancies images made by Legendre method (parametrized).

3.2.2 Gaussian Fields method

- To build a synthetic abundancies sample with the Gaussian Fields method just type the command:

  ```
  [abundanciesSample] = getAbundanciesSampleGaussianFields();
  ```

  that will create a sample of abundancies images with the default parameters (Figure 3.15). The "abundanceSample" will be a matrix of $256 \times 256 \times 5$ elements, being the first and second the spatial dimensions and the last the number of abundancies images.

  The parameters used by the Gaussian Fields method are:

  - ne: number of abundancies images in the sample (default: 5).
  - height, width: spatial dimensions of the abundancies images (defaults: 256,256).
  - type: type of Gaussian Fields used (default 1)
    - 1: Spheric
    - 2: Exponential
    - 3: Rational
    - 4: Mattern
  - theta1, theta2: parameters for the Gaussian Field type. It depends of each type of Gaussian Field. To see it in detail use the help command:

    ```
    help getAbundanciesSampleGaussianFields;
    ```

  Some uses:

  ```
  [abundanciesSample] =
  getAbundanciesSampleGaussianFields(4,256,256,1);
  ```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Spheric Gaussian Field with the default parameters (Figure 3.8).

```
[abundanciesSample] =
getAbundanciesSampleGaussianFields(4,256,256,1,50,0);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Spheric Gaussian Field with the parameters changed to $theta1 = 50$, $theta2$ is not used for this Gaussian Field type (Figure 3.9).

```
[abundanciesSample] =
getAbundanciesSampleGaussianFields(4,256,256,2);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Exponential Gaussian Field with the default parameters (Figure 3.10).

```
[abundanciesSample] =
getAbundanciesSampleGaussianFields(4,256,256,2,1,2);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Exponential Gaussian Field with the parameters changed to $theta1 = 1$ and $theta2 = 2$ (Figure 3.11).

```
[abundanciesSample] =
getAbundanciesSampleGuaussianFields(4,256,256,3);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Rational Gaussian Field with the default parameters (Figure 3.12).

```
[abundanciesSample] =
getAbundanciesSampleGaussianFields(4,256,256,3,1.2,0.85);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Rational Gaussian Field with the parameters changed to $theta1 = 1.2$ and $theta2 = 0.85$ (Figure 3.13).

```
[abundanciesSample] =
getAbundanciesSampleGuaussianFields(4,256,256,4);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Mattern Gaussian Field with the default parameters (Figure 3.14).

```
[abundanciesSample] =
getAbundanciesSampleGaussianFields(4,256,256,4,5,0.5);
```

that will create a sample of 4 abundancies images with 256 pixels-height and width, using the Mattern Gaussian Field with the parameters changed to $theta1 = 5$ and $theta2 = 0.5$ (Figure 3.15).

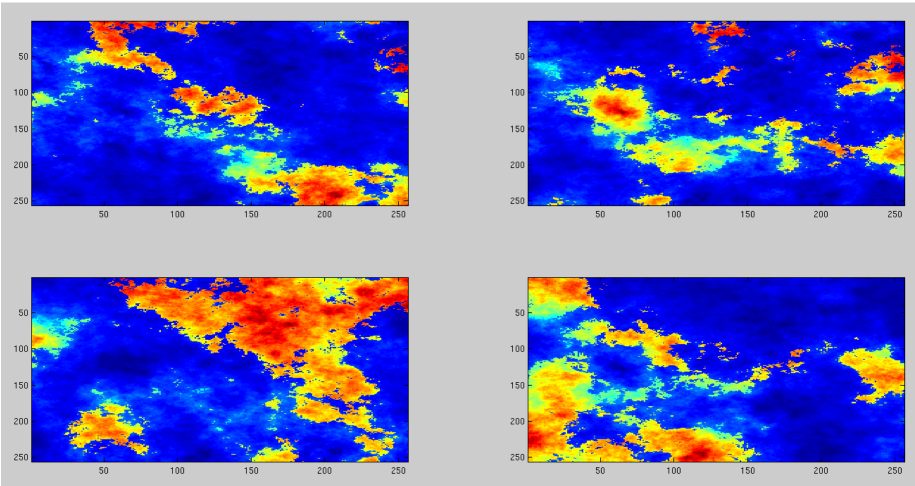

Figure 3.8: Sample of abundancies images made by Spheric Gaussian Fields method (default parameters).

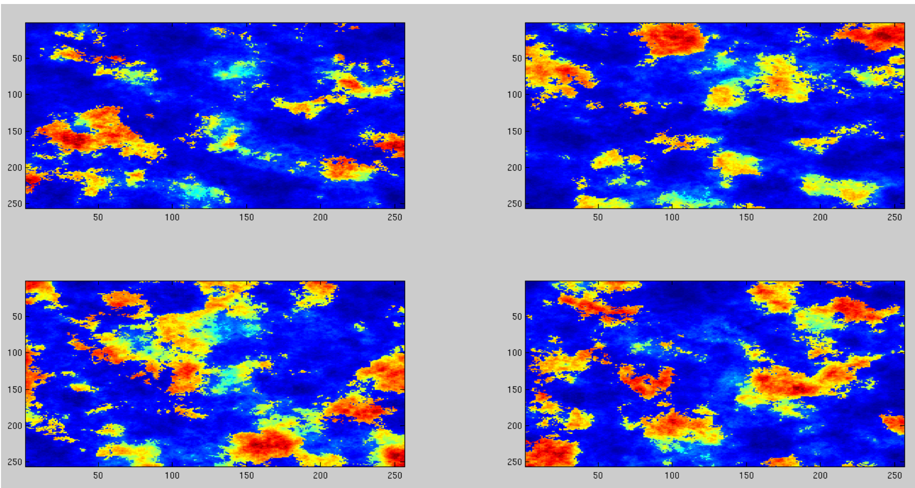

Figure 3.9: Sample of abundancies images made by Spheric Gaussian Fields method (parametrized).

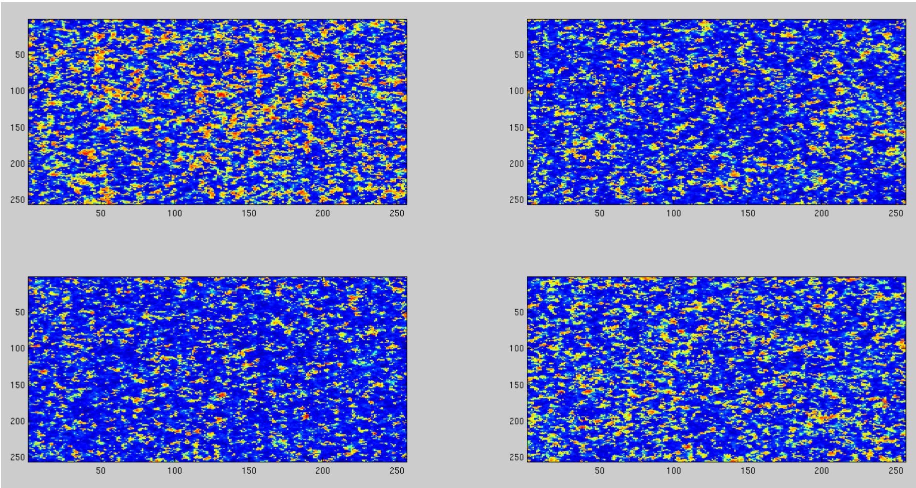

Figure 3.10: Sample of abundancies images made by Exponential Gaussian Fields method (default parameters).

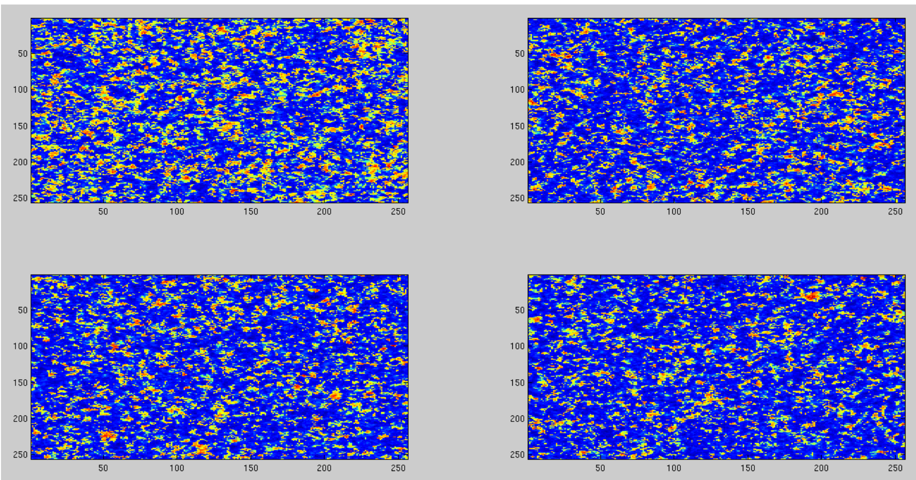

Figure 3.11: Sample of abundancies images made by Exponential Gaussian Fields method (parametrized).

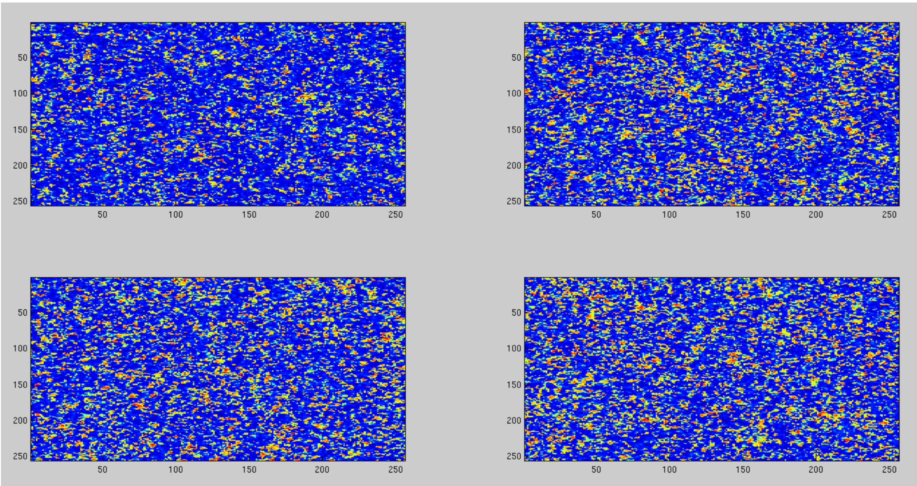

Figure 3.12: Sample of abundancies images made by Rational Gaussian Fields method (default parameters).

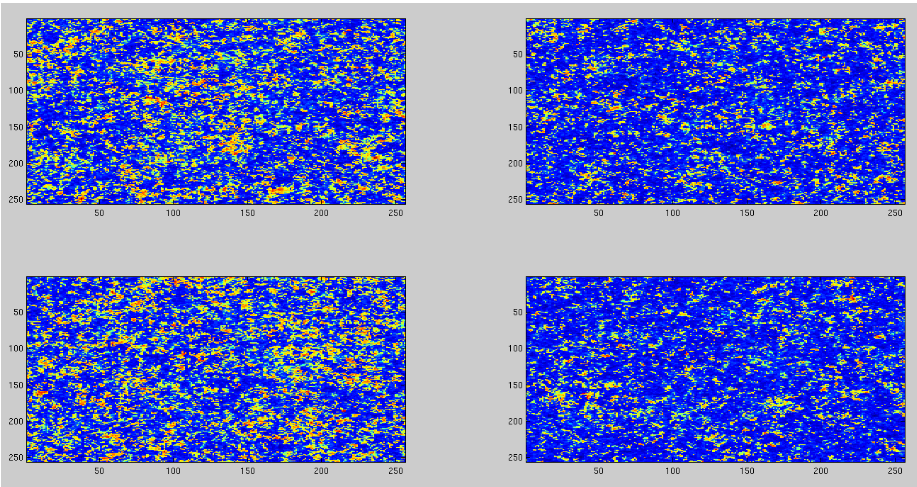

Figure 3.13: Sample of abundancies images made by Rational Gaussian Fields method (parametrized).

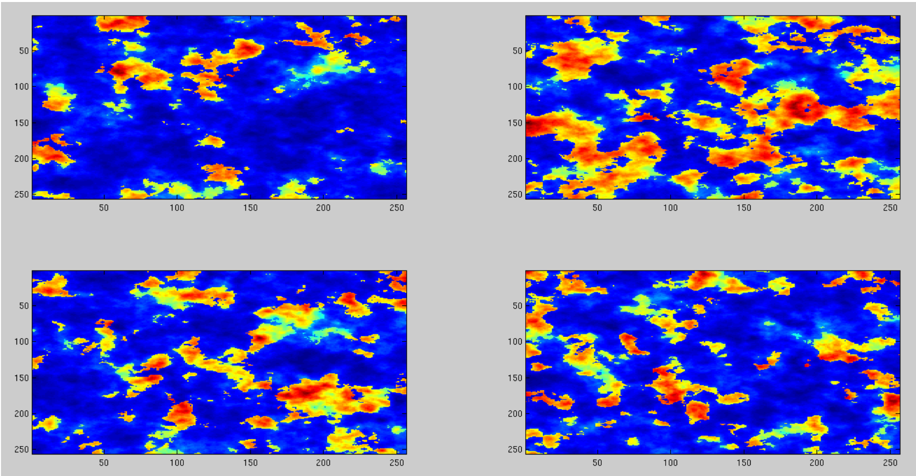

Figure 3.14: Sample of abundancies images made by Mattern Gaussian Fields method (default parameters).

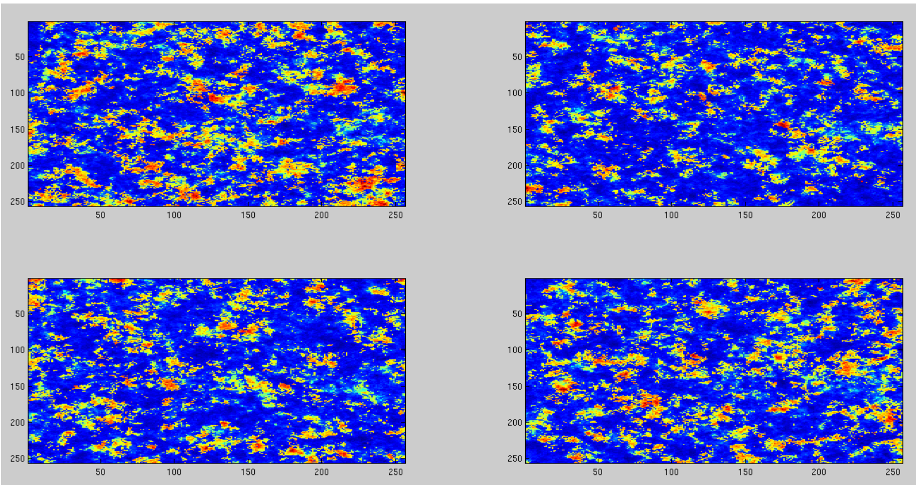

Figure 3.15: Sample of abundancies images made by Mattern Gaussian Fields method (parametrized).

- To build a collection of synthetic abundancies samples with the Gaussian Fields method just use the command:

```
[abundanciesCollection] = getAbundanciesCollectionGaussianFields(n);
```

that will create a collection of n Gaussian Fields abundancies samples with the default parameters. The “abundanciesCollection” will be a matrix of $n \times 256 \times 256 \times 5$ elements, being the first the sample index, the second and third the spatial dimensions of each sample and the last the number of abundancies images.

To control the parameters of the Gaussian Fields method, just put the parameters explained above after the n parameter. An example:

```
[abundanciesCollection] =
getAbundanciesCollectionGaussianFields(20,3,64,64,4,5,0.5);
```

will create a collection of twenty Mattern Gaussian Fields samples with 3 abundancies images each. In this case the “abundanciesCollection” will be a matrix of $20 \times 64 \times 64 \times 3$ elements, being the first the sample index, the second and third the spatial dimensions of each sample and the last the number of abundancies images.

3.3 Making hyperspectral synthetic images

Nota: In all the examples, the “endmembersReflectance” variable contains the reflectance values of the set of selected endmembers from the USGS library as explained in the “Loading endmembers” subsection.

3.3.1 Legendre method

- To buid a synthetic hyperspectral image by Legendre method just use the command:

```
[syntheticImage] = getHyperspectralSyntheticImageLegendre(endmembers);
```

where “endmembers” contains the reflectance values of the materials forming the hyperspectral image. This command will create a default abundancies sample by Legendre method to make the synthetic hyperspectral image. The resulting image is a matrix of $256 \times 256 \times p$ where the first and second are the default spatial dimensions of the image and the last is the number of spectral bands given by the endmembers set.

An example:

```
endmembers = endmembersReflectance(1:4,1:5:2151);
[syntheticImage] = getHyperspectralSyntheticImageLegendre(endmembers);
```

where the first four endmembers of the selected set are used to form the hyperspectral image.

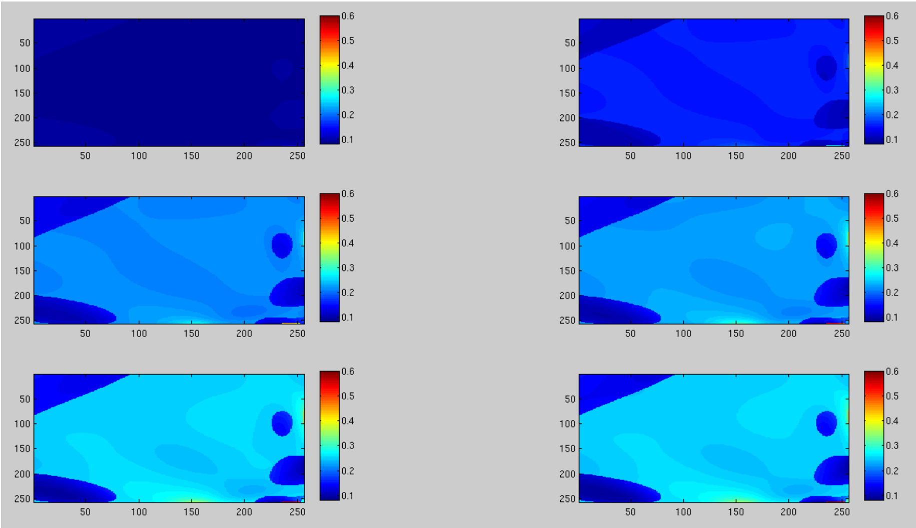

Figure 3.16: It shows some slices of the hyperspectral syntehic image cube made by Legendre method.

- To buid a synthetic hyperspectral image collection by Legendre method just use the command:

```
[syntheticImageCollection] =
getHyperspectralSyntheticImageCollectionLegendre(n,endmembers,ne);
```

where “n” is the number of images in the collection, “endmembers” contains a collection of reflectance values of endmembers, and “ne” is the number of endmembers forming each hyperspectral image. This command will create a default abundancies collection by Legendre method to make the synthetic hyperspectral image collection. The resulting image is a matrix of $n \times 256 \times 256 \times p$ where the first dimension is the index of images in the collection, the second and third are the default spatial dimensions of the image and the last is the number of spectral bands given by the endmembers set.

An example:

```
[syntheticImageCollection] =
getHyperspectralSyntheticImageCollectionLegendre(20,endmembers,4);
```

that will create a collection of twenty hyperspectral synthetic images by Legendre polynomies method, using for each image four different random endmembers of the full set of endmembers.

3.3.2 Gaussian Fields method

- To buid a synthetic hyperspectral image by Gaussian Fields method just use the command:

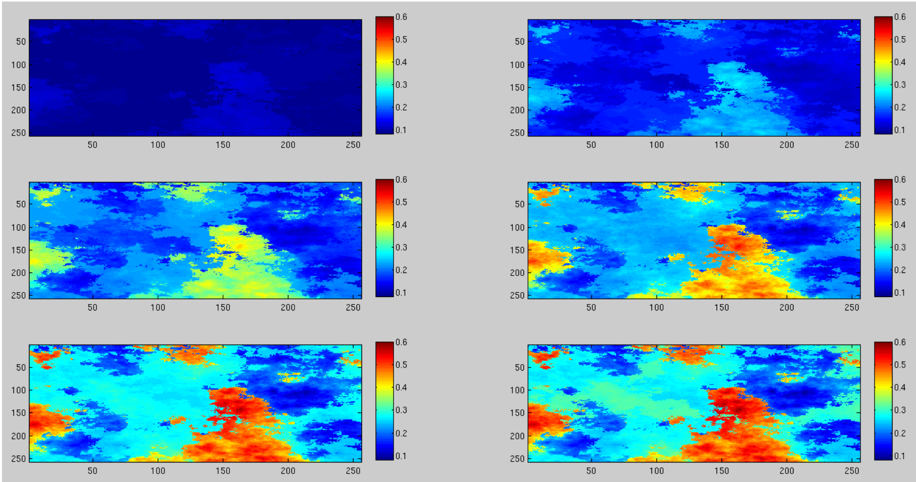

Figure 3.17: It shows some slices of the hyperspectral syntehic image cube made by Gaussian Fields method.

```
[syntheticImage] =
getHyperspectralSyntheticImageGaussianFields(endmembers);
```

where “endmembers” contains the reflectance values of the materials forming the hyperspectral image. This command will create a default abundancies sample by Gaussian Fields method to make the synthetic hyperspectral image. The resulting image is a matrix of $256 \times 256 \times p$ where the first and second are the default spatial dimensions of the image and the last is the number of spectral bands given by the endmembers set.

An example:

```
endmembers = endmembersreflectance(1:4,1:5:2151);
[syntheticImage] =
getHyperspectralSyntheticImageGaussianFields(endmembers);
```

where the first four endmembers of the selected set are used to form the hyperspectral image.

- To buid a synthetic hyperspectral image collection by Gaussian Fields method just use the command:

```
[syntheticImageCollection] =
getHyperspectralSyntheticImageCollectionGaussianFields(n,endmembers,ne);
```

where “n” is the number of images in the collection, “endmembers” contains a collection of reflectance values of endmembers, and “ne” is the number of endmembers forming each hyperspectral image. This command will create a default abundancies collection by Gaussian Fields method to make the synthetic hyperspectral image collection. The resulting image is a matrix of $n \times 256 \times 256 \times p$ where the first dimension is the index of images in the collection, the second and third are the default spatial dimensions of the image and the last is the number of spectral bands given by the endmembers set.

An example:

```
[syntheticImageCollection] =
getHyperspectralSyntheticImageCollectionGaussianFields(20,endmembers,4);
```

that will create a collection of twenty hyperspectral synthetic images by Gaussian Fields method, using for each image four different random endmembers of the full set of endmembers.

3.3.3 By prebuild abundancies

- To buid a synthetic hyperspectral image by the use of a prebuild abundancies sample use the command:

  ```
  [syntheticImage] = getHyperspectralSyntheticImage(endmembers,abundancies);
  ```

  where “endmembers” contains the reflectance values of the materials forming the hyperspectral image and “abundancies” is an abundancies sample. The resulting image is a matrix of $m \times n \times p$ where the first and second are the default spatial dimensions of the image given by the abundancies sample and the last is the number of spectral bands given by the endmembers set.

  An example:

  ```
  endmembers = endmembersreflectance(1:4,1:5:2151);
  [abundancies] = getAbundanciesSampleGaussianFields();
  [syntheticImage] =
  getHyperspectralSyntheticImage(endmembers,abundancies);
  ```

  where the first four endmembers of the selected set and a default Spheric Gaussian Field abundancies sample are used to form the hyperspectral image.

- To buid a synthetic hyperspectral image collection by the use of a prebuild abundancies sample use the command:

  ```
  [syntheticImageCollection] =
  getHyperspectralSyntheticImageCollection(n,endmembers,ne,abundanciesCollection);
  ```

  where “n” is the number of images in the collection, “endmembers” contains a collection of reflectance values of endmembers, “ne” is the number of endmembers forming each hyperspectral image and “abudnanciesCollection” is a prebuild collection of abundancies samples. The resulting image is a matrix of $n \times w \times h \times p$ where the first dimension is the index of images in the collection, the second and third are the spatial dimensions of the image given by the abundancies, and the last is the number of spectral bands given by the endmembers set.

  An example:

```
[abundanciesCollection] = getAbundanciesCollectionGaussianFields(20,4);
[syntheticImageCollection] =
getHyperspectralSyntheticImageCollection
(20,endmembers,4,abundanciesCollection);
```

that will create a collection of twenty hyperspectral synthetic images by Gaussian Fields method, using for each image four different random endmembers of the full set of endmembers.

Chapter 4

Synthesis Tool GUI

4.1 Synthesis Tool GUI code

The Synthesis Tool Graphical User Interface is included under the "gui" folder. This fodler contains three files:

- synthetic_run.fig: The figure design file used by the GUIDE Matlab Toolbox.
- synthetic_run.m: the ".m" file corresponding to the figrue file, containing the code.
- endmembers.mat: a ".mat" file containing variables related to the USGS endmembers library, needed by the interface to properly work.

4.2 Using the Synthesis Tool GUI

To run the Synthssis Tool GUI just type the command:

```
synthetic_run
```

An interface will appear (Figure 4.1). The interface has three panels: the Abundances, Images and Endmembers panels. Abudnances panel allows the user to select the way abundance images will be created. Images panel allows the user to select the amount of hyperspectral synthetic images to be created, as well as the spatial (lines and samples) dimensions of the images and the number of spectral signatures (endmembers) that are going to be used to build the image. This endmembers are randomly selected from the set of available endmembers choosed on the Endmembers panel. The sampling can be do with or without repetition.

The endmembers panel show the available spectral signatures from the USGS spectral library. This is indexed by the number of spectral bands (Bands field) and the type of materials (Dataset field). The user will add spectral signatures to form the set of endmembers from which select the endmembers to build each image. All the endmembers

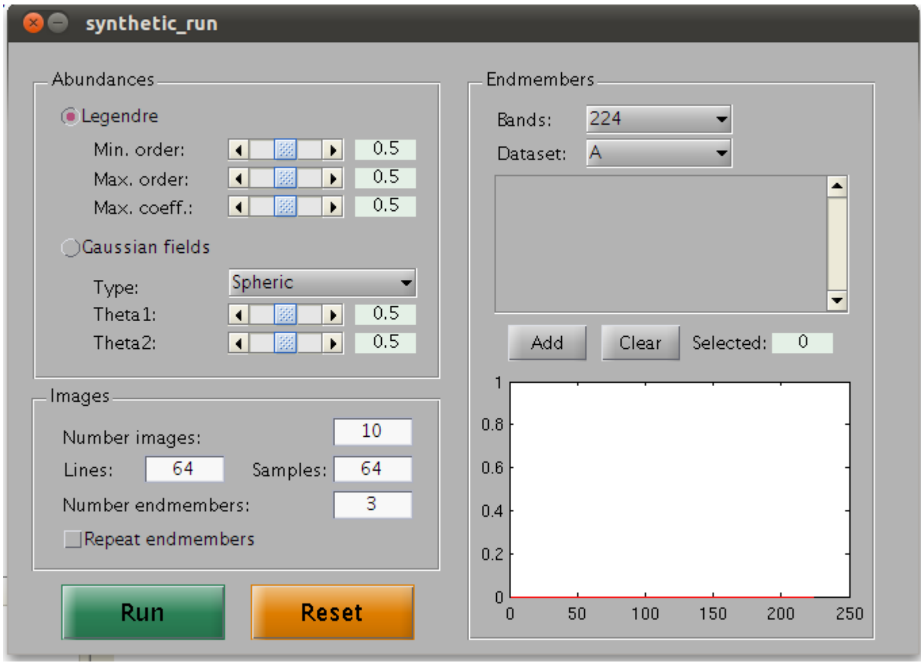

Figure 4.1: Synthetic Tool GUI

must have the same number of spectral bands, but they could be selected from different datasets. When one endmember is selected, its spectral signature will be show in red in the plotting figure (Figure 4.2). Once the selected endmember is added to the set it will be shown in blue, and the counter will be increased (Figure 4.3).

When you are ready to build the images, press the “Run” button. A window resuming the building process will be showed (Figure 4.4). If agree, the images will be created and saved in “hyper-N.mat” files under the current work folder (Figure 4.5). It will be created as many files as images. Each file contains the hyperspectral synthetic image, the abundance images and the endmembers used in the synthesis process.

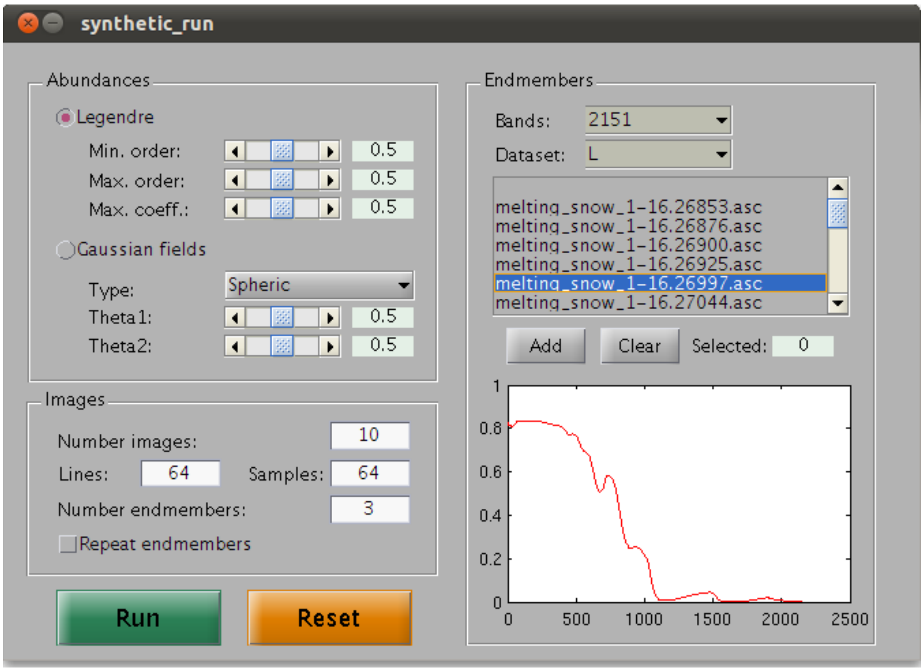

Figure 4.2: Selecting an endmember

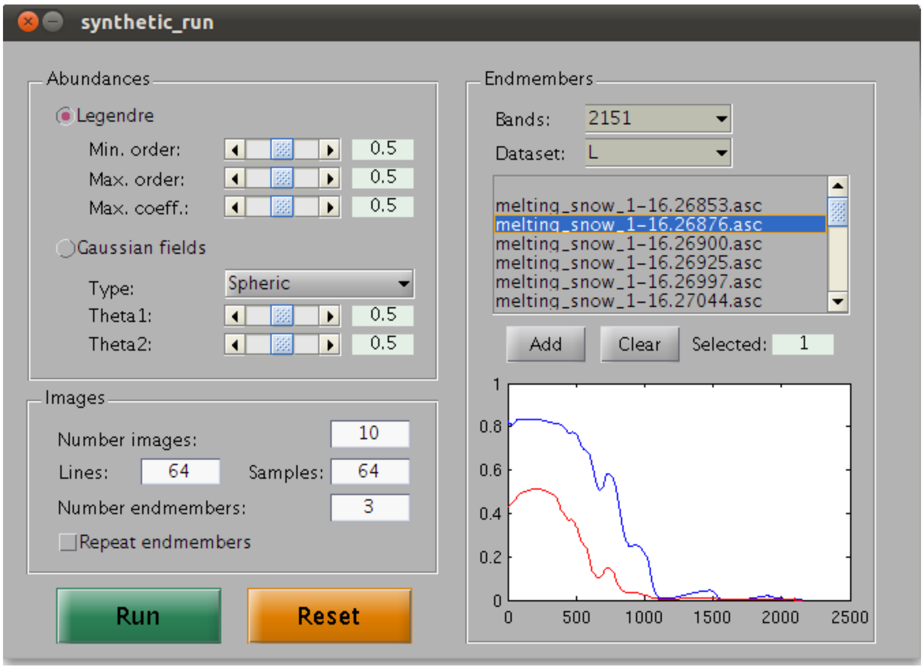

Figure 4.3: Adding an endmember

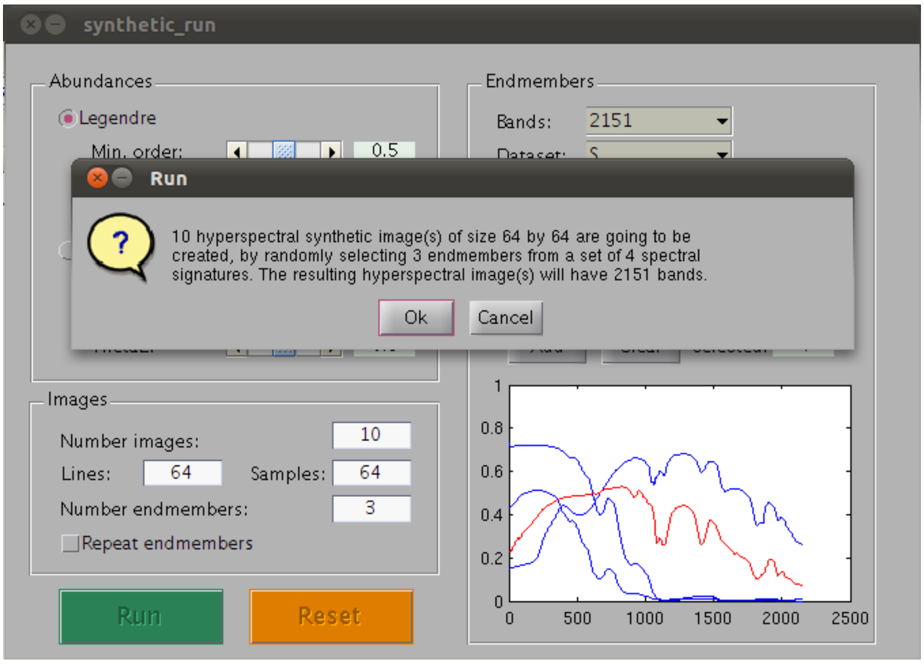

Figure 4.4: Running the synthesis process

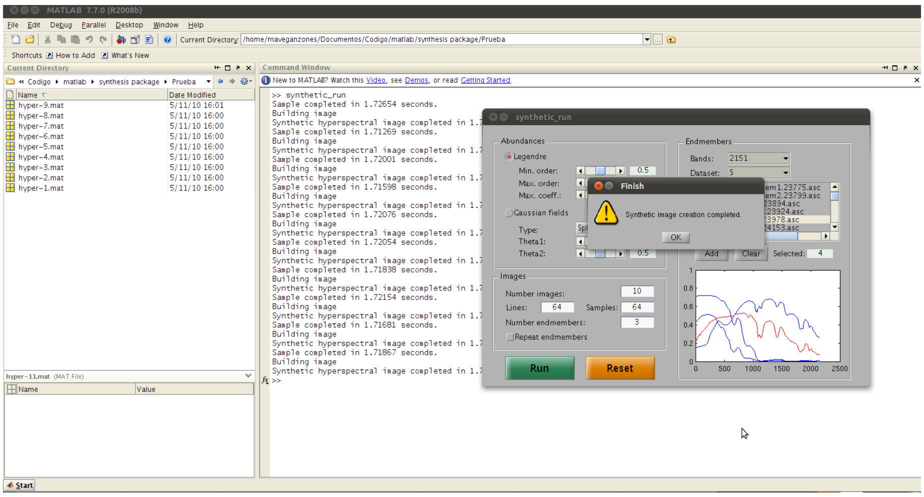

Figure 4.5: The synthetic image files creation completed