C: DIAGNOSTIC PRODUCTS FOR SURFACE REFLECTANCE IMAGES

Like *Frequently Asked Questions*, a question is posed, e.g., *C1. How Do I Make a Mask (MK) Raster?* Then, an answer is given[1] with comments and opinions. For cross referencing, each item is labeled, e.g., *C1*.

This tutorial deals with DIAG.sml, its uses, and its options.

DIAG.sml produces diagnostic products from SRFI rasters aided by a related pair of PVI and PBI rasters (see *B36*) and a matching binary mask raster, MK. For example, if the input rasters are SRFIBL, SRFIGL, SRFIRL, SRFINA, PVI, and PBI, then the output diagnostic rasters will be DBL, DGL, DRL, and DNA, respectively. In many cases, the input rasters for DIAG.sml were produced by SRFI.sml. However, the user may have used REPAIR_IMAGE.sml to fix the original image DNs or TERCOR.sml before using (or re-using) DIAG.sml on the resulting SRFI, PVI, and PBI rasters.

DIAG.sml also requires the existence of a binary MK raster that matches the extents, orientation, and cell sizes of the SRFI, PVI, and PBI rasters. You must produce the MK raster manually using TNTmips tools. This tutorial provides instructions about how you can produce MK. *Table A7* gives the multispectral (MS) two-letter band code that is used to identify each MS band.

Diagnostic products help you determine the SRFI spectral signature for two specific kinds of land cover: bare soil and dense vegetation. These are key inputs to subsequent analysis tools. With a well-selected set of parameters, DIAG.sml may produce other “pure-pixel” signatures of interest.

However, before you actually run DIAG.sml, you must make the MK raster. It focuses DIAG.sml on specific areas that are most likely to yield the desired sets of “pure” SRFI spectral signatures.

When you create the MK raster, it needs to define areas of generally level terrain. Alternatively, you can use TERCOR.sml to correct SRFI values for the effects of non-level terrain (see *D*). REPAIR_IMAGE.sml looks for image DNs that are equal to the fill-image DN value (usually 0) and reassigns these pixels to a DN value either equal to 1 or equal to the fill-value minus 1.

TERCOR.sml (see *FAQs_by_Jack_D.doc* or *FAQs_by_Jack_D.pdf*) requires that you:

- Have a digital elevation model (DEM) or a digital terrain model (DTM) that covers the same geographic extent as the SRFI rasters.

[1] Jack F. Paris, Ph.D., 2407 Maplewood Cir. E., Longmont, Colorado 80503 USA, *jparis37@msn.com* , 303-775-1195

- Have SRFI rasters that are accurately orthorectified. That is, the DEM or DTM pixels must be well registered with SRFI pixels.

Some MS scenes have essentially level terrain. In this special case, it is not necessary to process the SRFI data through TERCOR.sml before you use DIAG.sml.

In any case, you must make a MK raster that matches the SRFI rasters that you will be processing with DIAG.sml.

In Brief …

This tutorial discusses key SML functions and model concepts related to **DIAG.sml**. The list below is divided into two groups: one for the key SML functions and the other for key model concepts.

If you are interested in a particular topic below, please go directly to it.

Quick Guide to Using DIAG.sml …

If you are already familiar with SML functions and syntax … and you just want to Run DIAG.sml, this Quick Guide will help you.

BEFORE you run DIAG.sml …

- Run SRFI.sml. It produces the SRFI, PVI, and PBI rasters that DIAG.sml uses as input rasters.
- You may also run TERCOR.sml, which also produces a set of terrain-corrected SRFI rasters and a related pair of PVI and PBI rasters.
- From the vendor's metadata, note the following information items: SITE NAME, COLLECTION DATE, SUN ELEVATION ANGLE, SUN AZIMUTH ANGLE, and IMAGING SYSTEM.

AFTER you start the script, the script will ask you to provide or to accept specific information items via a series of **Popup Windows**, in the following order:

- **CONSOLE-WINDOW ADJUSTMENT:** Use your mouse to adjust the size and placement of the Console Window. You need to be able to view its contents as the script runs and prints data to it.
- **SITE-NAME ENTRY:** Type in the SITE NAME, e.g., Stockton, CA
- **IMAGER-NUMBER SELECTION:** From the list, select an Imager Number and type it, e.g., 4 for Landsat 7 ETM+.
- **SOIL-SEARCH PVI-WIDTH ENTRY:** Accept the default width, 100. If you believe that a wider search is necessary, increase this parameter.
- **SOIL-SEARCH PBI-MAX ENTRY:** Accept the default max, 900. If you believe that the soil is darker or light than this, input a different value.
- **VEG-SEARCH PVI-MIN ENTRY:** Accept the default min, 1400. If you believe that vegetation of interest to you has a different min value, input a different value.
- **PURE-SOIL-PIXEL SOIL-SD-MAX ENTRY:** Accept the default entry, 14. If you believe that the texture filter needs to be more tolerant of spatial variability, then increase this entry.
- **PURE-VEG-PIXEL VEG-SD-MAX ENTRY:** Accept the default entry. It is based on the Pure-Soil-Pixel Soil-SD Value that you selected in the last PopupNum Window.
- **PURE-PIXEL BOX-SIZE ENTRY:** Accept the default entry, 5. If you want to exclude more pixels, make this box-size larger, e.g., 7, 9, or 11. If you want to include more pixels, make this box-size smaller, e.g., 3.
- **SRFI, PVI, and PBI RASTERS** are input rasters.
- **D RASTERS** are output rasters. Put them in a new .RVC file called DIAG.

The script takes a few minutes (up to 15 minutes) depending on the size of the imager file being processed.

C1. How Do I Make the MK Raster?

You use two menu-driven processes in TNTmips, as follows:

Process 1: Spatial Data Editor.

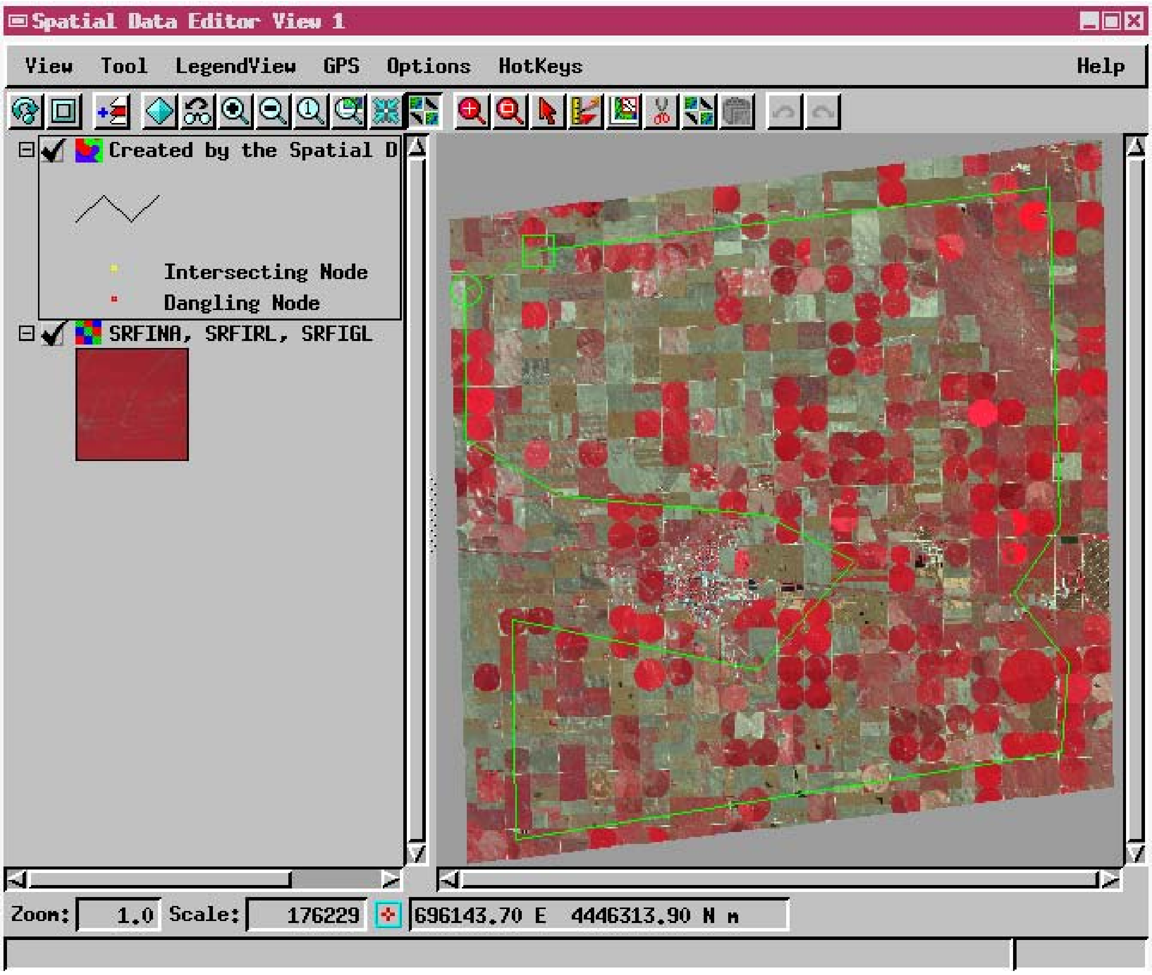
From the TNTmips main menu:

- Menu path: Edit > Spatial Data… The Spatial Data Editor control box appears with the Spatial Data Editor View 1 window.
- In the Spatial Data Editor control box, Select Reference. A pull-down list appears.
- Select Add Raster. Another pull-down list appears.
- Select Quick-Add RGB. The Select Objects control box appears.
- Navigate to the Project File that contains SRFI rasters. For Red, Green, and Blue, select SRFINA, SRFIRL, and SRFIGL, respectively. This RGB combination will be displayed in the Spatial Data Editor View 1 window. You should be able to recognize major kinds of land cover (and areas that were not imaged: black or transparent areas where null values exist). Dense vegetation is red. Senesced vegetation and bare soil will be greenish or grayish. Open water will be dark and bluish. Buildings in urban areas will be gray or white. Woodland and trees will be dark red. *Figure C1* shows how this display looks for a QuickBird (QB) MS scene collected over Yuma, CO, on July 2, 2003.

> The green line in this figure will be discussed soon. It does not exist until you draw a polygon over the Reference image. Polygons identify the areas that will eventually have a MK raster value of **1**. The scene in *Figure C1* is dominated by cropland, which has, of course, lots of dense vegetation and bare soils. In a different scene, you must make a MK polygon that encloses mostly dense vegetation and bare soil.

- In the Spatial Data Editor control box, Select File> New > Vector… . The New Object Values selection box appears.
- Click the OK button. A new Vector layer appears in control box and in the window of the Spatial Data Editor. Also, the Vector Tools – 2D-XY Polygonal tool box appears.
- In this tool box, Select Add Polygons (A). This is identified when you move your mouse cursor over it (third from the left in the top row). The Line/Polygon Edit Controls box appears.
- Make sure that the Stretch button is pushed **in** inside the Mode panel. Also, the Add End button should be pushed in inside the Operation panel.
- When you move the mouse cursor over the displayed Reference image, it changes to a pencil icon. You are about to use it to draw one or more polygons over the displayed image.

Figure C1: Color Infrared (CIR) Combination of SRFI Rasters



- In *Figure C1*, a simple polygon was drawn. It is not necessary that you be precise in drawing this polygon. This polygon was drawn to avoid urban areas. It also stayed well within the imaged area (to avoid areas having no data: the gray areas). And, it avoid non-level terrain, large open-water areas, and, cloudy areas.
- When you are happy with the polygon's boundaries, Click the Add button. The **polygon line turns black**. If you wish, you can draw and Add more polygons. But, one well-drawn polygon is sufficient.
- In the Spatial Data Editor control box, click the Tools icon button for the vector layer, select Validate from the drop-down list. Then, do the same for the Update Standard Attributes item in the list.
- In the Spatial Data Editor control box, select Save as… from the File menu. The Select Object selection box appears.
- Navigate to the Project File that contains the SRFI, PVI, and PBI rasters that you intend to process with DIAG.sml.

- Click the New Object… button (which is identified when you move your mouse cursor over it).
- Highlight the NewVector (default name) and change this to MK.
- Click the OK button on the New Object dialogue window. Click the OK button (close the time advice) on the Process Status Dialogue window.
- Click the red **X** in the Spatial Data Editor control box to close this tool.

Process 2: Vector to Raster Conversion.

From the TNTmips Main Menu:

- Menu path: Process > Convert button. A pull down list appears.
- Select Vector to Raster… . Two control windows and a view window appear.
- In the Vector to Raster Conversion control box, Click the Input Vector… button.
- Navigate to the Project File that contains the MK vector. Select it. The polygon(s) appear in the View window.
- You need to set all of the Input Parameters now as follows:
    - Lines Process: Set to None.
    - Polygons Process: Set to All.
    - Polygons Value: Set to All Same. Click on the yellow pencil icon and Specify… that this value be **1**. Do not allow this to keep its default value of **0**. Pixels inside of the polygon(s) need to have a binary raster value of **1**; outside areas will have a binary raster value of **0**.
    - Enable the Use reference raster button (push it in: to have an **x** on it).
    - Click the Input Raster… button.
    - Navigate to one of the SRFI rasters and select it. It appears (in gray tones) under the black vector polygon in the View window.
    - Now, Click the Run button.
    - Navigate to the Project File that contains the SRFI, PVI, and PBI rasters and take the default name: MK.
- After the process finishes, you will see the MK raster in **black** (**0**) and **white** (**1**) in the View Window. This tells you that you successfully made the MK raster! Congratulations!
- Click the Exit button. You are now ready to use DIAG.sml.

Look at the various statements in DIAG.sml before you Run it.

C2. How Do I Select the SRFI Rasters to be Processed?

In SRFI.sml, the user specified an imaging system by an imager number. Then, SRFI.sml used this selection to set the values for the Boolean variables called pBL … pMC. You will use the same method in DIAG.sml.

Value of Working with SRFI, PVI, and PBI as Input Rasters.

SRFI, PVI, and PBI could have been derived from any MS sensor. The value of working with calibrated rasters like SRFI, PVI, and PBI is that they are sensor independent! The numeric values in these rasters are given in a standardized scale that is consistent enough to be used in an absolute sense in subsequent processes.

In fact, the SRFI values and the related PVI and PBI values might have also been corrected for terrain effects by TERCOR.sml. They also might have been moved spatially, e.g., in an orthorectification process. All of these preprocessing operations are unimportant to the functioning of DIAG.sml.

C3. What is `wpvis`*?*

This is one of the "search-box" parameters. In the case, the search is done in spectral space (2-Space) not in geospatial space. Specifically, it is the width of the search box for bare soil pixels in terms of the value of PVI. Recall that PVI was designed to have a nominal value of 1000 on the Line of Bare Soils (refer to *B36*). Since PVI and PBI are calibrated indices, the true location of the Line of Bare Soils is near the line defined by PVI = 1000. The real Line of Bare Soils, therefore, is expected to be somewhere within a small range of PVI values around this nominal value of PVI. That "range" is the value of **wpvis**, which, in plain English, is the width (**w**) of PVI (**pvi**) values as related to the Line of Bare Soils (**s**).

Based on the user-specified value of **wpvis**, the values of the lower and upper limits of PVI for these two "edges" of the "search box" related to the location of the Line of Bare Soils are calculated, as shown in DIAG.sml as **pvi1s** and **pvi2s**, respectively.

The related PopupNum function uses a long prompt that is built on several **string** variables, **p1$ … p7$**. the default value (100), the minimum allowed value (2), the maximum allowed value (500), and the number of decimal places for this parameter (0).

C4. What are `pbi2s` *and* `pbi1s`*?*

The upper end of the bare-soil search box is defined in terms of PBI by **pbi2s**. From this, the lower end of the bare-soil search box in terms of PBI is defined by **pbi1s**, as indicated in the script. The bare-soil search box is limited in both the PBI and PVI so that clouds (very high values of PBI), urban and water (very low values of PVI with low PVI values) are excluded.

C5. What is **pvi1v**?

The lower end of the dense vegetation search box is defined in terms of PVI by **pvi1v**. There is no upper end of this "box." Recall that the tip of the Tasseled Cap (TC) distribution have high values of PVI. All other land-cover types and clouds have values of PVI that are well below the PVI for the dense-vegetation tip of the TC. This characteristic of the TC distribution makes it unnecessary to put limits on PBI to search for dense-vegetation pixels.

The related PopupNum function states the prompt ("Veg-Search PVI Min?"), the default value (1300), the minimum allowed value (900), the maximum allowed value (2500), and the number of decimal places for this parameter (0). 900 is specified for the minimum allowed value just in case the user wants to search all pixels that have a mixed soil-vegetation signature (and that are spatially-smooth pixels, as explained soon).

C6. What are **boxsize, sdmaxs,** and **sdmaxv**?

In the logic of DIAG.sml, a set of conditional tests determine whether or not a given pixel is:

- In the spatial area defined by the processing mask, MK" and either
    - In the search box related to bare soil PVI and PBI signatures or
    - In the search box related to dense vegetation PVI signatures.

But, there is one more condition test that the pixel must pass before being used to make non-zero values in the output diagnostic rasters. In plain English, it must be surrounded by other pixels that have PVI values similar to its PVI value. That is:

- It cannot be near a sharp boundary, where PVI changes across the boundary.
- Nor can it be in an area where the variability of PVI is large within a defined focal area.

The size of the focal area, in pixels, must be an odd integer. This is called the **boxsize**. DIAG.sml has logic that forces boxsize to be an odd integer (even if the user inputs an even number).

The SML function, FocalSD, returns the value of the standard deviation of the raster values that are within a focal area (boxsize by boxsize) for the indicated raster, e.g., PVI. The focal point is given by a lin and col position. The returned value for **pvisd** is then compared to either **sdmaxs** or to **sdmaxv** for the bare-soil pixel search or for the dense-vegetation pixel search, respectively.

Using two **sdmax** parameters accounts for scale differences for PVI values related to bare soil (near 1000) and those related to dense vegetation (near 2000). The user may choose values for **sdmaxs** and **sdmaxv** that are not equal to the suggested default values (if they are within a reasonable range). The author could have used Coefficient of Variation (cv) values, where cv = sdpvi * 100 / meanpvi. But, this would require an additional calculation (with the FocalMean function), which increases the run-time for the SML. The suggested upscaling of **sdmaxv** relative to **sdmaxs** accomplishes the goal of using a texture parameter like cv while avoiding another Focal function.

You may be wondering why the author did not have a texture test for variations in the PBI raster values? One goal of DIAG.sml is to find pixels associated with the Line of Bare Soils. If vegetation is present, then PVI values will likely vary spatially. If vegetation is not present, PVI values will not change much spatially. However, soil brightness (represented by PBI when PVI is near a value of 1000) can … and does … vary greatly. The author allows the spatial variation of soil brightness to be present so long as the variation due to vegetation is low. In the case of dense vegetation, the spatial variation of PVI also becomes low as vegetation becomes dense.

It is well known that changes in SRFI are small in the near infrared and in the RL band when vegetation is dense and rich with chlorophyll and high in leaf area index or biomass density. Thus, both PVI and PBI have small spatial standard deviations. But, it is necessary to test in only one of these, namely, PVI.

C7. How are Likely Bare-Soil and Dense-Vegetation Pixels Found?

The outside "for each" loop related to this question runs through all of the pixels in the MK rasters – one at a time (one **mk** value at a time).

If the pixel is inside of the mask, then it is True (has a value of 1). **mk** acts here as a Boolean variable.

If **mk** is True, the pixel's value of **pvi** is fetched from the PVI raster. **pvi** is tested against **pvi1s** and **pvi2s** to determine if the pixel is *possibly* inside of the bare-soil search box.

If so, then the pixel's value of **pbi** is fetched from the PBI raster. Then, **pbi** is tested against **pbi1s** and **pbi2s** to determine if the pixel is inside of the bare-soil search box. If this condition is met, then **pvisd** is calculated (using the FocalSD function) and then is tested against **sdmaxs**). If all of these bare-soil tests pass, the program fetches values of SRFI and assigns them to open D rasters at the same pixel location (as controlled by the Boolean **p** variables).

If **mk** is True, the pixel's value of **pvi** (already fetched from the PVI raster) is tested against **pvi1v** to determine if the pixel is inside of the dense-vegetation search "box."

If this condition is met, then **pvisd** is calculated (using the FocalSD function) and then is tested against **sdmaxv**). If all of these dense-vegetation tests pass, the program fetches values of SRFI and assigns them to open D rasters at the same pixel location (as controlled by the Boolean **p** variables).

C8. How do the D Rasters Compare to the SRFI Rasters?

If you open a CIR combination of RGB rasters using SRFINA , SRFIRL, and SRFIGL rasters, you will get an image that looks like *Figure C8a* (next page). It shows a community that has mixed land cover. This area is an "urban" area that was not excluded from the MK mask.

Figure C8b (next page) shows the same area as a CIR combination of RGB rasters using DNA, DRL, and DGL rasters. It is clear that DIAG.sml did a good job in finding pure pixels that are likely to be either bare soil or dense vegetation.

Figure C8a. CIR Based on SRFINA, SRFIRL & SRFIGL.

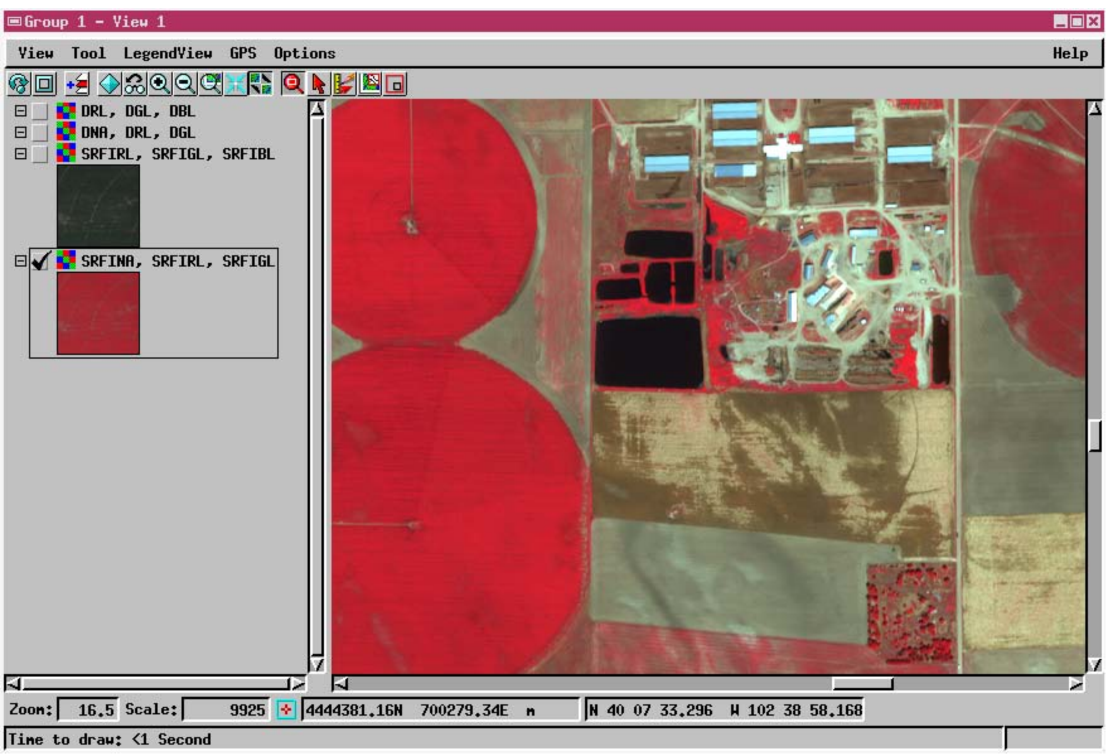
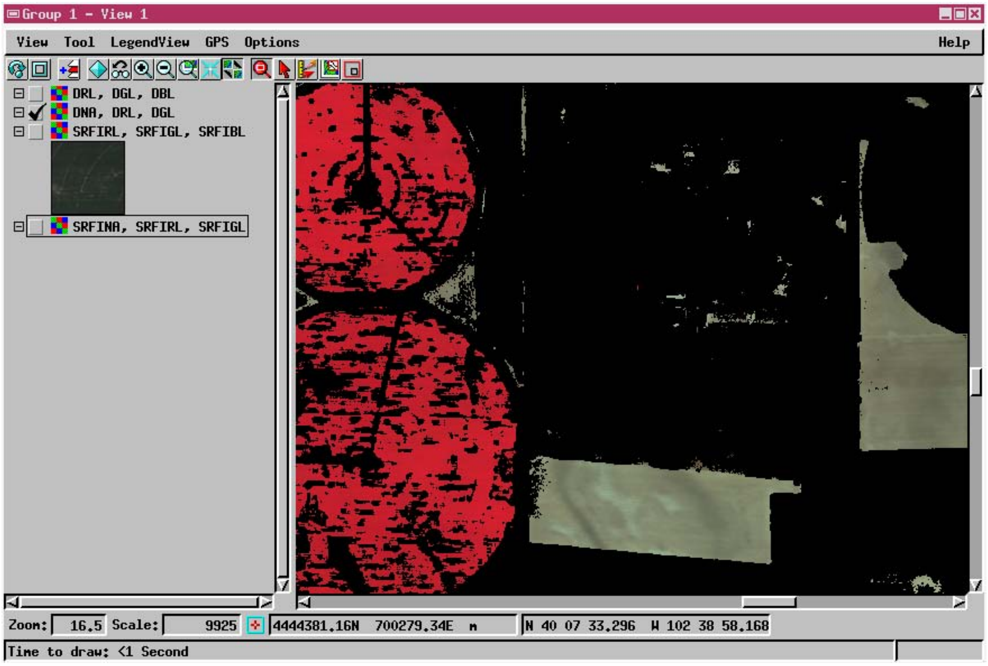
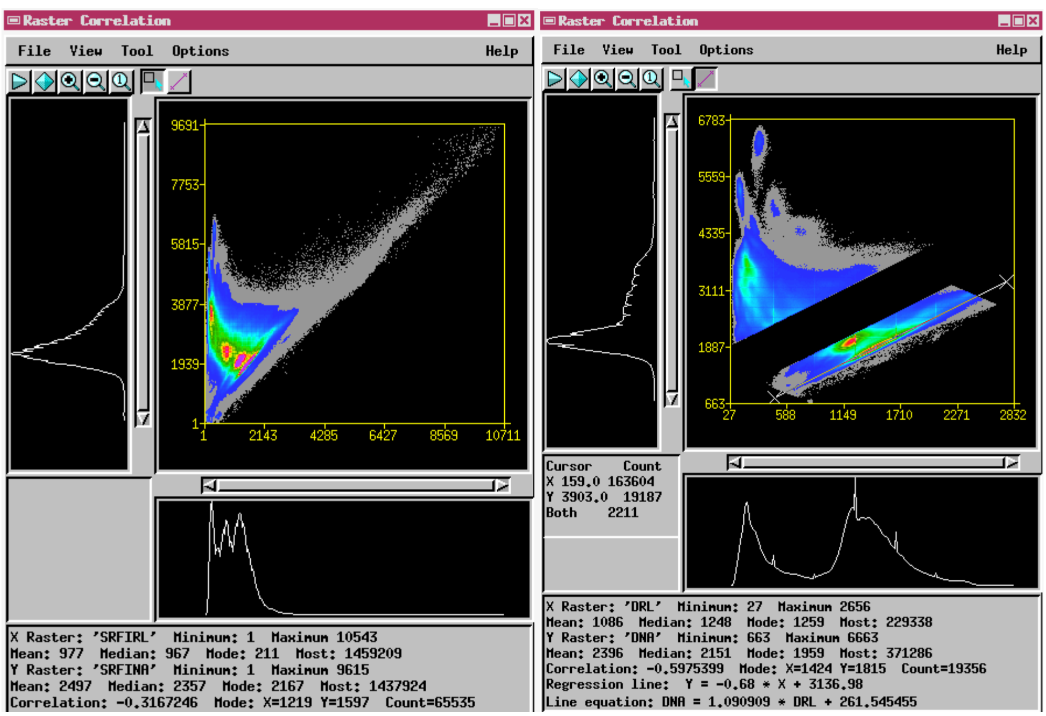


Figure C8b. CIR Based on DNA, DRL & DGL Rasters.



A comparison of the scatterplot of SRFINA vs. SRFIRL (in ***Figure B8c***) and the scatterplot of DNA vs. DRL (in *Figure B8d*) also shows the excellent performance of the DIAG.sml script.

Figures C8c and C8d. Scatterplots SRFINA vs. SRFIRL (left) Compared to DNA vs. DRL (right).



The Line of Bare Soils is easy to find in *Figure C8d* (right). Also, candidates for dense vegetation are easy to find in this scatterplot. The author changed the Palette color associated with density level 1 from blue to gray. Note that the box-search algorithm combined with the texture-search algorithm excludes mixed pixels that have SRFI values that cannot be either bare soil or dense vegetation.

The Line of Bare Soils in ***Figure C8d*** is easily found. In this case, the author found this line to be defined by:

DNA = 261 + 1.090909 * DRL **(C8a)**

Since DNA and DRL values are the same as SRFINA and SRFIRL, the equation for the Line of Bare Soils in SRFINA vs. SRFIRL feature space is also:

SRFINA = 261 + 1.090909 * SRFIRL **(C8b)**

You should keep the **D** rasters so that you can extract other features from them. The non-zero valued pixels in the **D** rasters are not affected much by spatial mixing nor by other types of land cover that may not be of interest to you, e.g., open water, urban materials, mixed vegetation and soil, clouds, and shadows.

For example, if you are interested in obtaining the pure spectral end members of a mixture of corn and background soil, you can get the required SRFI values for these scene features. On example of this is given in ***Table C8*** below:

Table C8. SRFI [and Standardized Reflectance Factors (SRF, in %)**] for Three End Members of a Mixture of Dense Corn and Bare Soil.**

Pure Material	**SRFIBL** (SRFBL %)	**SRFIGL** (SRFGL %)	**SRFIRL** (SRFRL %)	**SRFINA** (SRFNA %)
Dense Corn	**264** (2.64%)	**382** (3.82%)	**159** (2.98%)	**3903** (34.2%)
Bright Bare Soil	**1036** (10.36%)	**1241** (12.41%)	**1742** (17.42%)	**2146** (21.46%)
Dark Bare Soil	**749** (7.49%)	**845** (8.45%)	**1112** (11.12%)	**1482** (14.82%)

These end-member spectra for dense corn and bare soil could be used as critical spectral parameters in a linear unmixing model or in a non-linear unmixing model in the quantitative analysis of pixels distributed over a given corn field.

All Vegetation Indices (VIs) represent a solution to a linear or non-linear mixing situation involving variable background soils and dense foreground vegetation. This theme will be explored in greater depth in subsequent SMLs that the author will write with the goal of producing VIs from SRFI values.

REFERENCES

Lillesand, T. M., R. W. Kiefer, and J. W. Chipman, 2004: *Remote Sensing and Image Interpretation*, 5th Ed., John Wiley & Sons, Inc., Hoboken, NJ, 763 pp.