# README file for the Global Precipitation Measurement (GPM) Validation Network (VN) Geometry-Match Data Visualization Software

## Overview

The GPM VN software packages for visualization and statistical analysis of geometrically-matched (mapped to common 3-D spatial volumes) TRMM Precipitation Radar (PR) and scanning Ground Radar (GR) data exists as a collection of Interactive Data Language (IDL) procedures and functions. The GPM VN prototype is being developed to assess the first-order differences between TRMM PR data and (primarily) the U.S. national network of operational S-band weather radars (Weather Service Radar-1988 Doppler, or WSR-88D) operated by NOAA/National Weather Service. The core GPM satellite will carry instrumentation similar to TRMM, and GPM data processing and products will mirror those of TRMM. The VN prototype supports GPM pre-launch algorithm development and future ground validation procedures.

Two analysis and display programs are provided in this software package. The first is the VN Statistical Analysis and Display Program, named **z_comparisons_driver**. Z_comparisons_driver reads geometry-matched PR and GR data and computes and reports PR-GR reflectivity difference statistics; displays vertical profiles, histograms, and scatter plots of PR and GR reflectivity; and displays elevation sequences of GR and PR reflectivity data as Plan Position Indicator (PPI) images. The second program is the Cross Section Analysis program, named **cross_sections_driver**. Cross_sections_driver displays vertical cross sections of geometry-matched PR and GR data along a PR scan; a matching vertical cross section of PR-GR reflectivity difference; and (optionally) a matching cross section of full-vertical-resolution PR corrected reflectivity from the original TRMM 2A-25 data product.

Both programs take previously-prepared geometry-match data files in netCDF format as their primary data input. The PR-GR geometry-matching program, POLAR2PR, is the IDL procedure which processes TRMM PR and coincident ground radar (GR, also called GV) data into a common set of spatial data volumes using a geometry-matching algorithm, and outputs the geo-match data to these netCDF files. The format and content of the netCDF files, and procedures to gain online access to the netCDF files already generated for the GPM Validation Network data set, are described in detail in the *GPM GVS Validation Network Product Data User's Guide*, available from the GPM GV web site (http://gpm.gsfc.nasa.gov/groundvalidation.html). The IDL source code for POLAR2PR is provided in a separate software delivery, available on the NASA/GSFC Open Source web site via the NASA/GSFC Innovative Partnerships Program Office (http://opensource.gsfc.nasa.gov). *If using VN geometry-match netCDF files from the GPM ftp site, then it is not necessary to obtain and run POLAR2PR to produce input for the visualization routines.*

## Source Code and Precompiled Binaries

IDL source code and precompiled binaries (IDL SAVE files), User's Manuals, and test data for **z_comparisons_driver** and **cross_sections_driver** are contained in the unix tar file, ***GPM_VN_Vis_Tools.tar***. See the Tar File Listing section below for a description and listing of this file's contents.

## Installation and Configuration

In the descriptions to follow, a basic knowledge of unix/linux shell commands, configuration of IDL, and compilation and running of IDL procedures is assumed.

1) Locate the secondary unix tar file ***GPM_VN_Vis_Tools.tar*** that was extracted, along with this README file, from the primary tar file downloaded from the GSFC Open Source web site, and copy or move it to the desired target directory where the software and test data are to be installed.

   NOTE: The precompiled SAVE files, the User's Manuals, the IDL source code, and the test data will reside in four new subdirectories created under the current working directory from which the ***GPM_VN_Vis_Tools.tar*** file's contents will restored in step 2:

   `bin/` – precompiled IDL SAVE files
   `doc/` – User's Manuals
   `src/` – IDL source code
   `testdata/` – sample test data

   If a subdirectory with one of these names is already present in the target install directory, consider renaming the conflicting directory, or installing the VN code in another directory. Be sure that IDL is configured to Search Subdirectories under the `src` directory where the source code will reside.

2) Restore the contents of the tar file, using a utility program or by running the following unix/linux command from the command line, making sure that the current working directory is the target install directory, and the tar file is in this directory:

   ```
   tar -xvf GPM_VN_Vis_Tools.tar
   ```

   Verify that the four new subdirectories are created, and that they are populated with the files listed at the end of this README file.

3) Make note of the target directory where the tar file has been restored. This directory path will be needed to configure the control files provided for running the test data cases. For the sake of the examples to follow, assume that the tar file

***GPM_VN_Vis_Tools.tar*** has been restored on the IDL host machine under the target directory `'/home/stormy/vntools'`, and the resulting restored files and directories are left as-is.

4) Once the ***GPM_VN_Vis_Tools.tar*** has been restored, this tar file can be deleted. If necessary, it can always be extracted again from the primary tar file obtained from the GSFC Open Source web site.

## Test Data

Inputs to **z_comparisons_driver** include geometry-match netCDF data files and a plain-text control file specifying the formal IDL parameters that provide program control and data file location (path/name) information to the **z_comparisons_driver** procedure.

Inputs to **cross_sections_driver** include geometry-match netCDF data files, and (optionally) two of the files used to generate the geometry-matched data file: the TRMM PR 2A-25 product file, and GR volume scan data in a Universal Format (UF). Other required inputs are a plain-text control file specifying the formal IDL parameters providing program control and data file location (path/name) information to the **cross_sections_driver** procedure. The TRMM PR 2A-25 and GR Universal Format products are optional, and **cross_sections_driver** can run successfully with only the geometry-match netCDF data files. See the User's Manual for the Cross Section Display Program for instructions on running the procedure with and without 2A-25 and UF data.

A full set of input test data and control files for both programs are provided in the tar file. The test data files are located under the `testdata` directory the in target directory where the tar file was restored. The sample control file for **z_comparisons_driver** is named `stats_analysis.ctl`, and the sample control file for **cross_sections_driver** is named `cross_sections.ctl`. Review the User's Guides in the `/doc` directory for details of the control parameters required by each program, and how they are defined in the control files. Also, refer to the <u>Notes on the `testdata` directory structure</u> section at the end of this document for details related to editing the control file parameter values to work with the test data provided with this code distribution. ***The control files must be edited to reflect local installed paths to the test data files before the programs can be run successfully.***

## Compiling and running the **z_comparisons_driver** and **cross_sections_driver** procedures

It is not necessary to compile **z_comparisons_driver** and **cross_sections_driver** in order to run them, since both programs (IDL procedures) are provided as precompiled binaries in the form of IDL 'SAVE' files, named `z_comparisons_driver.sav` and `cross_sections_driver.sav`. These Save files are located in the `/bin` subdirectory under the target directory. Detailed directions for configuring the necessary input parameters in the control files, configuring IDL, and restoring and running the programs are given in the User's Manuals for each program, located in the `/doc` directory of the code installation.

Instructions for compiling and running the procedures are given below only for informational purposes, for users who are familiar enough with IDL to compile and run the source code from scratch, or for cases where the source code is to be modified.

Following installation of IDL source code and suitable test data, the IDL procedures **z_comparisons_driver** and **cross_sections_driver** may be compiled and run. The top-level source code file for **z_comparisons_driver** is `z_comparisons_driver.pro` and for **cross_sections_driver** is `cross_sections_driver.pro`, and each is located in the `/src` subdirectory.

First, start IDL or the IDL Development Environment (IDLDE). If this has not already been done, then add the full pathname to the `/src` directory of this code distribution (e.g., `/home/stormy/vntools/src` in our example) to the IDL path, and make sure IDL is configured to Search Subdirectories under this file path. At the IDL> prompt, enter the following command:

```
IDL> .compile z_comparisons_driver.pro
```

If the IDL path has been properly set up, the **z_comparisons_driver** procedure should compile with no errors and display the diagnostic output:

```
% Compiled module: PARSE_PDF_SCA_PARMS.
% Compiled module: Z_COMPARISONS_DRIVER.
```

This is a partial compilation of the full set of code involved in the **z_comparisons_driver** procedure. The remaining procedures and functions will be compiled as they are called, when **z_comparisons_driver** is run.

Then run **z_comparisons_driver** by simply entering the name of the procedure at the IDL> prompt:

```
IDL> z_comparisons_driver
```

The procedure should then start, and a user interface should appear to allow selection of the control file. As needed, edit the Directory field in the user interface to point to the `/home/stormy/vntools/testdata` directory (in our example, where the target directory is `/home/stormy/vntools`), press the Filter button until the control files appear in the file select list, and select the `stats_analysis.ctl` file.

Similarly, to compile and run the **cross_sections_driver** procedure, enter the following command at the IDL> prompt:

```
IDL> .compile cross_sections_driver.pro
```

If the IDL path has been properly set up, the **cross_sections_driver** procedure should compile with no errors and display the diagnostic output:

```
% Compiled module: PARSE_XSECT_PARMS.
% Compiled module: CROSS_SECTIONS_DRIVER.
```

Then run **cross_sections_driver** by simply entering the name of the procedure at the IDL> prompt:

```
IDL> cross_sections_driver
```

Again, the procedure should then start, and a user interface should appear to allow selection of the control file. As needed, edit the Directory field in the user interface to point to the `testdata` directory, and press the Filter button until the control files appear in the file select list, and select the `cross_sections.ctl` file.

## Constraints

***A copy of IDL, running under Unix/Linux or Mac OS X, is required to compile and/or run the software.*** *The z_comparisons_driver procedure will compile and run in IDL demo mode (an unlicensed IDL version), but will not be allowed to produce an output Postscript/Adobe PDF data file.* Refer to the User's Manuals for additional input data, software, and runtime environment constraints that pertain to the **z_comparisons_driver** and **cross_sections_driver** procedures. The procedures have been tested in a Linux environment with IDL Versions 6.3 and 7.1, and in Mac OS X in IDL 7.0. ***The software will <u>not</u> run successfully in the Windows operating system.***

## File listing for the tar file ***GPM_VN_Vis_Tools.tar***

The following files and directory structures are contained in the tar file ***GPM_VN_Vis_Tools.tar***. These directories and files will be created under the directory from which is restored by running the "`tar –xvf GPM_VN_Vis_Tools.tar`" command or a file unpacking utility.

```
bin/
      cross_sections_driver.sav
      geo_match_z_pdf_profile_ppi_bb_prox_sca_ps.sav
      pr_and_geo_match_x_sections.sav
      z_comparisons_driver.sav

doc/
      GEO_MATCH_CROSS_SECTIONS_USER_GUIDE.pdf
      STATISTICAL_ANALYSIS_PROGRAM_USER_GUIDE.pdf

src/
      calc_geo_pr_gv_meandiffs_wght_idx.pro
      cos_and_tan_of_pr_angle.pro
      cross_sections_driver.pro
      environs.inc
      find_alt_filename.pro
      find_pr_products.pro
      geo_match_nc_structs.inc
      geo_match_z_pdf_profile_ppi_bb_prox_sca_ps.pro
      get_rsl_radar.pro
      get_site_specific_z_volume.pro
      get_uf_pathname.pro
      grid_def.inc
      grid_nc_structs.inc
      loadcolortable.pro
      loop_pr_gv_gvpolar_ppis.pro
      plot_geo_match_xsections.pro
      plot_pr_xsection.pro
      plot_scatter_by_bb_prox.pro
      plot_scatter_by_bb_prox_ps.pro
      plot_sweep_2_zbuf.pro
      pr_and_geo_match_x_sections.pro
      pr_params.inc
      read_2a25_ppi.pro
      read_geo_match_netcdf.pro
      read_pr_2a25_fields.pro
      s_band_to_ku_band.pro
      standard_error.pro
      uncomp_file.pro
      vn_colorbar.pro
      z_comparisons_driver.pro
      rsl_in_idl/
            CHANGES.txt
            COPYING.txt
            get_dimensions.pro
            getmetaobject.pro
            get_parmnames_hdf.pro
            gv_radar_site_info.data
```

```
is_compressed.pro
jul2cal.pro
monthname.pro
nsig_v1_define_structs.pro
nsig_v2_define_structs.pro
plot_range_rings2.pro
README.txt
rsl_add_volume.pro
rsl_adjust_coord.pro
rsl_anyformat_to_radar.pro
rsl_basename.pro
rsl_changefield.pro
rsl_colorbar.pro
rsl_fix_time.pro
rsl_get_azm_from_sweep.pro
rsl_get_date_from_filename.pro
rsl_get_fields.pro
rsl_get_groundr_and_h.pro
rsl_get_gr_slantr_h.pro
rsl_get_numvos.pro
rsl_get_radar_site_info.pro
rsl_get_range_from_sweep.pro
rsl_get_ray_from_sweep.pro
rsl_get_ray.pro
rsl_get_site_from_filename.pro
rsl_get_slantr_and_elev.pro
rsl_get_slantr_and_h.pro
rsl_get_sweep.pro
rsl_get_uf_last_sweepnum.pro
rsl_get_volume.pro
rsl_hdf_to_radar.pro
rsl_lassen_to_radar.pro
rsl_loadcolortable.pro
rsl_new_radar.pro
rsl_new_ray.pro
rsl_new_sweep.pro
rsl_new_volume.pro
rsl_nsig_to_radar.pro
rsl_open_radar_file.pro
rsl_plotrhi.pro
rsl_plotsweep2pixmap.pro
rsl_plotsweep_from_radar.pro
rsl_plotsweep.pro
rsl_radar_to_uf_gzip.pro
rsl_radar_to_uf.pro
rsl_read_lassen_file.pro
rsl_select_sweeps.pro
rsl_uf_to_radar.pro
rsl_uncompress.pro
rsl_vslice.pro
rsl_which_struct.pro
rsl_wsr88d_to_radar.pro
VERSION.txt
wsr88d_get_site_info.pro
wsr88d_locations.dat
wsr88d_read_tape_header.pro
ymd.pro
```

```
            doc/
                  hdf_to_radar.html
                  Install.html
                  radar_header.html
                  radarstruct.html
                  ray_header.html
                  raystruct.html
                  rsl_add_volume.html
                  rsl_anyformat_to_radar.html
                  rsl_changefield.html
                  rsl_get_fields.html
                  rsl_get_groundr_and_h.html
                  rsl_get_gr_slantr_h.html
                  rsl_get_ray_from_sweep.html
                  rsl_get_ray.html
                  rsl_get_slantr_and_elev.html
                  rsl_get_slantr_and_h.html
                  rsl_get_sweep.html
                  rsl_get_volume.html
                  RSL_in_IDL.html
                  rsl_plotrhi.html
                  rsl_plotsweep_from_radar.html
                  rsl_plotsweep.html
                  rsl_radar_to_uf_gzip.html
                  rsl_radar_to_uf.html
                  rsl_routines.html
                  rsl_select_sweeps.html
                  rsl_structures.html
                  rsl_vslice.html
                  rsl_which_struct.html
                  sweep_header.html
                  sweepstruct.html
                  volume_header.html
                  volumestruct.html
                  whatsnew.html

testdata/
      cross_sections.ctl
      stats_analysis.ctl
      netcdf/
            geo_match/
                  GRtoPR.KAMX.060808.49749.nc.gz
      prsubsets/
            2A25/
                  2A25.060808.49749.6.sub-GPMGV1.hdf.gz
      gv_radar/
            KAMX/
                  1CUF/
                        2006/
                              0808/
                                    060808.21.MIAM.4.2006.uf.gz
```

Notes on the **`src`** directory:

1. The files under the src directory are necessary only if compiling the **z_comparisons_driver** and **cross_sections_driver** procedures, and/or if modifying the source code. As previously noted, the procedures are provided as precompiled, binary IDL Save files located under the bin directory. The Save files can be restored and run directly in IDL without the need to compile the procedures.
2. If the POLAR2PR source code distribution has been installed, then there will be duplicate instances of the `rsl_in_idl` source code on the system. As long as one of these instances is within the IDL search path, then the duplicate instance can be deleted, but it should not be necessary to do so.
3. One RSL routine (`rsl_get_sweep.pro`) in the tar file is a modified version of the baseline RSL in IDL code set. It has been modified to deal with WSR-88D "split cut" volume scans, where there are duplicate sweeps at the same elevation angle. Otherwise, all the files under the directory `rsl_in_idl/` are complete and unmodified from the RSL in IDL distribution.
4. The file `src/loadcolortable.pro` substitutes for the baseline RSL procedure contained in `rsl_loadcolortable.pro`, and has been customized to meet the PPI display capabilities of the various VN programs.

Notes on the **`testdata`** directory structure and control file parameters:

1. The geometry-match netCDF files used by both procedures are located in the subdirectory `testdata/netCDF/geo_match`, reflecting the structure of the operational GPM Validation Network (VN) database. These files are stored under a default directory `(/data/netCDF/geo_match)` in the VN database system. This default directory is defined inside the **cross_sections_driver** program, and should be overridden by specifying the local directory path (see step 3 of Installation, above) to the netCDF files as the value for the **NCPATH** keyword parameter in the control file. For example, if the target directory used to restore the tar file was `/home/stormy/vntools`, then the default root directory should be overridden by specifying the complete local directory path to the netCDF files as the value for the **NCPATH** keyword parameter in the `stats_analysis.ctl` and `cross_sections.ctl` control files, as shown:

   ```
   NCPATH=/home/stormy/vntools/testdata/netcdf/geo_match
   ```

   Any additional geometry-match netCDF files obtained from the VN ftp site or created by POLAR2PR should be placed in the directory pointed to be NCPATH, or NCPATH should be edited to point to the directory where the additional files are located.

2. The 2A25 test data file type used by **cross_sections_driver** is stored under a `/2A25` subdirectory under `testdata/prsubsets`, reflecting the structure of the operational GPM Validation Network (VN) database. It holds TRMM PR data files of the type indicated by the file and directory names (e.g., TRMM 2A-25 products in

the `2A25` files/directories). The operation VN data set also contains files and subdirectories for TRMM PR product types 1C21 and 2B31. ***It is assumed that these subdirectories 1C21, 2A25, and 2B31 are all located under a common root directory*** (`/data/prsubsets` in the VN database system). This default root directory is defined inside the **cross_sections_driver** program, and should be overridden by specifying the ***partial*** target directory path (see step 3 of Installation, above) to the `prsubsets` root directory as the value for the **PRPATH** keyword parameter in the `cross_sections.ctl` control file, as shown:

```
PRPATH=/home/stormy/vntools/testdata/prsubsets
```

Any additional 2A-25 files obtained from the VN ftp site or other PR data sources should be placed in the `/2A25` subdirectory under PRPATH, or PRPATH should be edited to point to the directory where the additional files are located.

3. The ground radar data files in Universal Format (UF) used by **cross_sections_driver** are stored in a site- and date-specific directory tree of the mandatory structure:

   ```
   station/filetype/year/monthday
   ```

   in the tar file, and in the operational VN database. These individual directory trees for each ground radar station are stored under a fixed, common root directory (`/data/gv_radar/finalQC_in`) in the VN database system. This default directory is defined inside the **cross_sections_driver** program, and should be overridden by specifying the local root directory path as the value for the **UFPATH** keyword parameter in the control file. For example, if the target directory used to restore the tar file was `/home/stormy/vntools`, then the full path specification for the ground radar data file provided as test data is:

   ```
   /home/stormy/vntools/testdata/gv_radar/KAMX/1CUF/2006/0808/
   ```

   In this case, the control file `cross_sections.ctl` for **cross_sections_driver** must be edited to specify the common root path value (the part of the path preceding the site ID 'KAMX') for the **UFPATH** parameter as shown:

   ```
   UFPATH=/home/stormy/vntools/testdata/gv_radar
   ```

   Any additional UF or 1CUF files obtained from the VN ftp site or other GR data sources should be placed in site- and date-specific directory structures of form `station/filetype/year/monthday` under UFPATH, or UFPATH should be edited to point to the directory where the additional directory structures and their UF files are located.