Polar Allsky Image Merging (PAIM) Tool

This document is a brief user's guide to the an IDL tool for merging allsky images from multiple polar cap stations onto a common geomagnetic grid. Since documentation is usually the last task performed during software design, we acknowledge that this guide is preliminary and we hope that your own hands-on experience with the PAIM code will provide a much better learning environment. An attempt has been made to internally document each IDL procedure (or at least the major ones) to aid the user. Before we describe the general capabilities of the tool, a few comments are in order:

(1) **Required Platform** This tool has been designed to operate in a UNIX environment which will support IDL version 3.5 and which is capable of rendering 900 by 900 PIXEL color images with eight bits of color per PIXEL.

(2) **Hardcopy Requirements** The PAIM code requires that a Postscript-compatable laser printer is available for making greyscale hardcopies. The exact syntax required to access your local printer should be inserted in the module *plot_image2.pro*, at line 183. This line defines a string, *printcom = 'lp' + fname*, which assumes that the laser printer can be accessed with a simple *lp* command. Simply replace this *lp* command with the appropriate command for your system. Color Postscript renderings are written to hard disk with filenames of the form *polar.cps*. If you are lucky enough to have access to a color printer, then you can send these files out after the IDL session is through. We have found that the best color hard copies and transparencies are made with dye sublimation type color printers.

(3) **Required Input Image Files** The PAIM tool operates on binary images which are subject to several assumptions. First of all, image arrays are assumed to be photometrically processed, that is, image data numbers can be ultimately related to Rayleighs. Overlapping images are averaged where required. The PAIM tool can be used with raw images but some attempt should be made to equalize the dynamic range of the individual images, lest data from overly bright images should dominate averaged PIXELs. Image arrays are assumed to be in the normal 'looking up' orientation, with East counterclockwise from North. It is additionally assumed that input images have a 1:1 aspect ratio. Thus, the allsky portion of images appear as circles rather than ovals. It is assumed that PIXELs outside of the circular image area have been set to a data number of zero. Lastly, it is assumed that any effects from lens distortion have been taken out. For such corrected images, the PIXEL radius from the center of the field of view (FOV) is linearly related to true elevation angle. Figure 1 shows typical input images from six polar sites.

General Description

The master program for the PAIM tool is an IDL procedure called *plot_image2.pro*. This program reads date, wavelength, image intensity calibration and UT data from a simple ASCII file called *images.dat*. This data file also contains the filename and image attributes for individual allsky images. Once these data are read, the master program creates a polar geomagnetic grid with the procedure *polarplot3.pro* that shows the UT-specific orientation of artic land masses. The program then creates geomagnetic look-up arrays that relate every point in the allsky FOV with a corresponding magnetic latitude and magnetic local time using a procedure called *allsky_mag1.pro*. Mapping is accomplished by smoothing the images over these geomagnetic coordinates. Thus, to run the tool, one need only modify the ASCII file *images.dat* to reflect the parameters of images to be merged and then type *plot_image2* at the IDL prompt.

On our SGI Indigo^2 workstation, the procedure requires approximately six minutes of 90% CPU access to map each image. Please be prepared for a slightly longer wait if you are using an older SPARC station, or if you are competeing with the unwashed masses for CPU time. Figure 2 shows the merged image obtained from running *plot_image2* with the images shown in Figure 1.

During the running of the PAIM tool, a window is opened up on the workstation to show the intermediate images as they are mapped. At the conclusion of the run, a 900 by 900 image array is saved to hard disk corresponding to the final composite image (*polar.img*). At this point, the user is given the option of saving the output array in any of three additional forms: a (1) black and white Postscript file (*polar.ps*), (2) color Postscript file (*polar.cps*), or (3) color GIF file (*polar.gif*). The specific color table used is loaded from an ASCII file called *good.cm*. If the user wishes to load one of the number IDL resident color tables, simply replace line 62 of procedure *plot_image2.pro* with the line: *loadct*, XX, where XX is the number of the lookup table to be loaded.

The input ASCII data file *images.dat* must be correctly structured for the PAIM tool to work. Because of its importance, a description of each required input variable is provided as follows:

Note: (a) through (g) are global variables.

(a) **Date**: A string variable for the UT date.

(b) **Wavelength**: A string variable denoting the emission wavelength in Å.

(c) **Image Rayleighs per data number**: A floating point variable used to label the maximum intensity shown on the output color bar. For raw image mapping, this variable should be set to one and the color bar interpreted as raw data numbers.

(d) **UT hours**: An integer variable for UT hours.

(e) **UT minutes**: An integer variable for UT minutes.

(f) **UT seconds**: An integer variable for UT seconds.

(g) **Number of images**: An integer variable for the total number of individual images to be merged.

Note: (h) through (s) must be specified for each image.

(h) **Image filename**: A string variable corresponding to the filename of the input byte array. This variable can include path information if the image in question resides in another directory.

(i) **Image byte array X dimension**: An integer variable specifying the horizontal dimension of the input image array.

(j) **Image byte array Y dimension**: An integer variable specifying the vertical dimension of the input image array.

(k) **Center field of view X pos**: An integer variable specifying the X coordinate value at the center of the circular image.

(l) **Center field of view X pos**: An integer variable specifying the Y coordinate value at the center of the circular image.

(m) **PIXEL radius from CFOV to im edge**: An integer varable specifying the radius in PIXEL units from the center of the image to the edge of the usable image.

(n) **Zenith distance at edge**: A floating point variable specifying the zenith distance corresponding to the edge of the usable image (in degrees). Allsky images are usually restricted to a maximum zenith distance of 75.0 degrees due to the gross distortion of image shapes at the edge of the FOV.

(o) **Azimuth at top of image**: A floating point variable specifying the azimuth angle associated with the very top center of the circular image.

(p) **Imager altitude**: A floating point variable specifying the geodetic altitude of the allsky imager.

(q) **Imager North geo latitude**: A floating point variable specifying the geodetic latitude of the allsky imager.

(r) **Imager East geo longitude**: A floating point variable specifying the geographic longitude of the allsky imager.

(s) **Centroid emission height**: A floating point variable specifying the geodetic altitude to which the emission is ascribed to (in km). Typical values are 250-350 km for 6300 Å and 110 km for 4278 Å.

A sample run of this PAIM tool has been prepared for inspection. Six fake images have been created with different image attributes and orientations. They are show in Figure 1. The specific data for each of these images has been preloaded into the file *images.dat*. Thus, one may simply type *plot_image2* at the IDL prompt and, approximately thirty minutes later, an image corresponding to Figure 2 will have been created. For users who demand instant gratification, just type *display_input* to see a summary of the input images, and then type *display_output* to see the resultant merged image. If the xview utility running on your computer the same files can be examined outside of IDL by typing *xv input.gif* and *xv output.gif* at the UNIX prompt.

A few words of caution are in order. Firstly, avoid modifying the text fields in *images.dat* as they are precisely positioned. If you want to create your own *images.dat* file copy the supplied version to another file name for safe keeping. Also, be sure to change the polar.* file names to something else unless you want to write over your beautiful images. I realize that these precautions could be avoided with clever software, but I am a bit too busy to work out the procedures. Good Luck.

Rick Doe
SRI International
30 August, 1994