afrl_bcd_overlap.pro

D. R. Mizuno

Institute for Scientific Research, Boston College, Chestnut Hill, MA, 02467

ABSTRACT

afrl_bcd_overlap.pro is an IDL routine for adjusting scalar levels of an ensemble of overlapping Spitzer BCDs so that they are collectively self-consistent. The usage of this software is presented. This document pertains to 17 November 2008 version of the program.

1. Introduction

Overlapping BCDs can have differing "DC" levels in the overlapping regions, due to various causes: BCDs containing bright sources may have "droop" error; BCDs in overlapping AORs may have differing Zodiacal levels, or errors in Zodiacal correction; and BCDs at the start of scan legs can have anomalous background levels. The solution is to add a scalar offset to each BCD, to minimize in the least-squares sense the level differences in the overlapping regions.

The MOPEX package from the SSC includes an overlap correction facility, but it has two major shortcomings: its use is practical only for a limited total number of BCDs (at most a few tens of thousands), and it offers only an "exact" overlap solution which is subject to a spurious ramping effect in the resulting offsets. The overlap correction presented here uses sparse-matrix operations, which allow for up to about 10^6 BCDs on current systems, and features an adjustable damping effect in the solution, which corrects local level deviations but generally preserves global background levels, at the cost of some residual overlap level differences. See Mizuno et al. (2008, PASP 120, 1028) for a discussion of the overlap correction procedure.

This routine should work with any IRAC or MIPS BCD data, although it has been extensively used and tested only on MIPS 24μm data. It has been used with Sun/Unix and Linux systems. It has not been used on PC versions of IDL.

2. Usage

The program is called `afrl_bcd_overlap.pro`. You will also need read access to the routines `linbcg_idl2.pro`, `return_masked_pixels.pro`, `return_valid_pixels.pro`, `apply_q.pro`, `strip_dir.pro`, and the standard astrolib routines.

The basic calling sequence in IDL is:

> `afrl_bcd_overlap, infiles`

`infiles` is an array (vector) of BCD FITS file names. The BCD ensemble must be contained in a region for which the maximum angular span is less that 180°. "Islands" of overlapping BCDs are OK; each "island" just gets a separate solution. The BCDs in the ensemble must all have the same array dimensions.

The remaining options are set with keywords. The ones that usually need to be set are:

`MASK_FILES`= Array (vector) of BCD mask FITS file names, corresponding one-to-one with the `infiles` list. If `MASK_FILES` is not set, all non-NaN pixels will be used in the overlap comparison (apart from other keyword specifications).

`BAD_BITS`= Array of bit numbers that flag bad pixels in the BCD masks. Default is bits 2 through 14. Or...
`BAD_BIT_CODE`= Bad bits represented as short-int value.

`OUT_DIR`= Output directory for corrected BCDs. Or...
`OUTFILES`= Array of output file names, corresponding one-to-one with `infiles`. If both `OUT_DIR` and `OUTFILES` are set, the `OUTFILES` file names will be written to the `OUT_DIR` directory. If neither are set, the routine will run but generate no output files. The routine will overwrite existing files so make sure that the output options are set appropriately.

`MAX_FLUX`= Set to maximum pixel brightness to use for overlap comparison. Generally, you will want to omit localized regions of very bright intensity, as the overlap comparisons are not reliable for very high intensity levels. There is no default value.

`OUTLIER_THRESH`= This is an option for "deweighting" outlier BCDs so that they float to match their neighbors without affecting the overall solution. BCDs whose median absolute overlap differences with their neighbors are above this value will be thus deweighted. Set to perhaps 10 times the "typical" overlap difference between BCDs (there shouldn't be more than a few percent deweighted BCDs). There is no default value.

`N_ITERATIONS`= This is the basic operational control for the damping effect. The default is 1 and gives a moderately damped result, which should be satisfactory if there are no large-scale level differences present (e.g. between entire AORs). For increasingly large values, the solution approaches the "exact" overlap solution (and its potential problems). Set to the smallest value that produces an acceptable result (so some trial-and-error may be necessary).

Other keyword options of occasional utility:

`WEIGHT_THRESH`= Normally all overlaps are weighted equally in the solution, and this usually gives the best results. For localized regions of very bright intensity, however, sometimes better results are obtained if the overlaps are weighted by overlap area. If there are correction errors in bright regions, try setting this keyword to a level just below the brightness of the problem area. Overlaps above that level will be weighted by area, and those below will be given uniform weighting. (Also, you could try setting MAX_FLUX to omit the bright pixels entirely from the overlap comparison, but then you might get no correction at all for some BCDs.) No default value.

STIFFNESS= This is the formal parameter for controlling the damping effect. The default is 0.05 and should generally be left at that value. In principle, smaller values approach the "exact" solution but in reality the solution machinery becomes unstable for small values of the stiffness, so use higher values of N_ITERATIONS to decrease the damping effect. You can set STIFFNESS to higher values to increase the damping effect but note that above about 0.5 there is very little correction applied at all.

MIN_OVERLAP= Set as the minimum number of pixels in an overlap region to include the overlap in the solution. Default is the number of pixels in 1 row in the BCD arrays.

SOURCE_FILE= Sometimes single BCDs will have differing background levels "above" and "below" a bright point source, due, e.g., to jailbar effects. There is a mechanism to split BCDs containing specified bright sources into sections, each of which obtains a separate correction in the solution. This is triggered by specifying SOURCE_FILE, which is set as the name of a text file containing such sources. The file contains 5 columns of free-format info. The columns are:
RA Dec Flux Galactic_lon Galactic_lat
The coordinate values are in decimal degrees, and only the RA and Dec values are used. The sources are mapped into the BCD arrays, and if a source falls on the array, the array is split into above-and-below sections for the overlap correction. Multiple sources on a single array are accomodated, but a minimum of 16 rows is required between sources (otherwise the "upper" source is ignored.)

OVERLAPS_PER_BCD= Set as the typical number of BCDs each BCD overlaps with. The default is 60. An exact value isn't necessary but the routine will run somewhat more efficiently if an approximate value is supplied.

3. Notes

Running time: The processing time depends on the array size and how the BCDs are distributed in the mapped region (i.e. number of overlaps per BCD). For the MIPS 24μm array (128×128), and 30-60 overlaps per BCD, on current Sun systems the processing time is about one hour per 10,000 BCDs. This should very roughly scale with array size (in total pixels) and overlaps per BCD. Note that the processing time is only weakly dependent on the setting of N_ITERATIONS; N_ITERATIONS=20, for example, will only increase the time by a few percent.

Memory requirements: This also depends on array size and BCD distribution. An approximate formula is:

$$\begin{aligned} \text{Memory (bytes)} \;\approx\;& 10\times(\text{total \# of BCD/BCD overlaps}) + \\ & 10 \times (\text{mean overlaps per BCD}) \times \\ & (\text{region span in BCD widths}) \times (\text{\# of pixels in BCD}) \end{aligned} \tag{1}$$

For oblong mapped regions, take the shorter axis as the region span.

Anomalous conditions: Regions of very bright extended emission are typically the most problematic for the overlap correction. Responsivity errors of a few percent in very bright regions can cause severe overlap correction discrepancies in more quiescent adjacent regions, and in some circumstances can cause the solution mechanism to fail. Careful setting of the `MAX_FLUX` and `WEIGHT_THRESH` parameters can mitigate these problem areas.

If the `OUTLIER_THRESH` parameter is set too low, it can cause entire sections of overlapping BCDs to be "deweighted." This can also cause problems for the solution mechanism, and will usually give a bad solution to such sections even if it converges properly. If `OUTLIER_THRESH` is set low enough that *all* BCDs are deweighted (or if an "island" of disconnected BCDs is entirely deweighted), this causes a singularity condition in the system of equations representing the corrections, and the solution mechanism will go haywire.