

MOPEX FAQ

By David Makovoz

03/01/06

Mosaicing – mosaic.pl

Q. What do I need to start using *mosaic.pl*?

A. A set of images with their filenames listed in a file.

Q. Are there any restrictions on the size of the mosaic due to the projection approximations?

A. The projection used here is exact, it involves no approximations. The only requirement is that the final mosaic can not span more then one hemisphere.

Q. What kinds of images can be processed?

A. Images in the FITS format. The following keywords are required:

BITPIX, NAXIS, NAXIS1, NAXIS2, CRVAL1, CRVAL2, CRPIX1, CRPIX2,
CTYPE1, CTYPE2.

Either the CD matrix or CDELTA1, CDELTA2, CROTA2 keywords are also required.

Q. How fast is the processing?

A. The slowest step is image interpolation. On a 500MHz Sunblade with 512 MB of memory it takes ~20sec to process a 256x256 image.

Q. Are there any restrictions on the mosaic pixel size?

A. There are no restrictions on the mosaic pixel size. If the mosaic pixel size is so small that you don't have enough computer memory to coadd all the interpolated images at once, you should split the mosaic into several tiles. In order to do so a suggested tile size is specified in the namelist:

```
&MOSAICCOADDIN  
  TILEMAX_X = 1000,  
  TILEMAX_Y = 1000,  
&END
```

Q. Can I do drizzle interpolation?

A. Yes. The drizzle option is set in the namelist:

```
&MOSAICINTIN
  # INTERP_METHOD = 1(default), 2(Drizzle),3(Grid),
  INTERP_METHOD = 2,
  DRIZ_FAC = 0.8,
&END
```

Q. When do I use `Grid` option for image interpolation?

A. Interpolation `Grid` option is used when a quick look mosaic is needed. It can also be used for background matching task (*overlap.pl*). The gain in speed is up to 10 times as fast as scheme one without much loss in the quality of the interpolated image. The price you pay is the fidelity of the interpolated images. It should not be used for photometry purposes. It also should not be used while doing outlier rejection. Don't forget to set `GRID_RATIO` in the namelist.

Q. Which modules should be turned on in the namelist for various outlier rejection options?

A. For single frame outlier rejection the following should be set to 1:

```
run_medfilter = 1
run_detect_radhit = 1
```

For multiframe outlier rejection the following should be set to 1:

```
run_mosaic_outlier = 1
USE_OUTLIER_FOR_RMASK = 1
```

For dual outlier rejection the following should be set to 1:

```
run_medfilter = 1
run_detect_outlier = 1
run_mosaic_proj = 1
run_mosaic_dual_outlier = 1
run_level = 1
USE_DUAL_OUTLIER_FOR_RMASK = 1
```

For multiframe box outlier rejection the following should be set to 1:

```
run_mosaic_box_outlier = 1
USE_BOX_OUTLIER_FOR_RMASK = 1
```

Interpolated images are required for multiframe and dual outlier rejection.

The following module should be run once before running `mosaic_rmask`

```
run_mosaic_covg = 1
```

Q. How do I set the pixel size of the mosaic?

A. You can set the mosaic pixel size directly using the following keywords in the namelist:

```
MOSAIC_PIXEL_SIZE_X = -0.0003
MOSAIC_PIXEL_SIZE_Y =  0.0003
```

`MOSAIC_PIXEL_SIZE_X` is negative since `CDELTA1` is negative. Alternatively, you can set the ratio of the input image pixel size to the mosaic pixel size using the following keywords in the namelist:

```
MOSAIC_PIXEL_RATIO_X = 2
MOSAIC_PIXEL_RATIO_Y = 2
```

Q. I am confused. In some places it says that I should use `THRESH_OPTION = 2` for the *mosaic_outlier* sigma computation option, because it is more aggressive and that `THRESH_OPTION = 1` doesn't work. Yet, in some other places it says that I should use `THRESH_OPTION = 1` because it is the robust option. Which one is it?

A. Both statements are true, but they apply to different versions of MOPEX. The older versions up to and including `mopex_093005` have the version of *mosaic_outlier* for which one should use `THRESH_OPTION = 2`, as the only option that actually detects outliers. However, this option is too aggressive, and starting with `mopex_030106` and on, `THRESH_OPTION = 1` was modified to use the median and median absolute deviation. This results in more reliable outlier detection with fewer false alarms. This option should be used for all the consecutive versions of MOPEX.

Q. I got the following error message:

```
/scr/mopex/bin/mosaic_coadd -n cdf/mosaic.nl -o
output_mosaic/Coadd_drizzle1/coadd -u
output_mosaic/Interp_drizzle1/interp_SigmaList.txt -f
output_mosaic/fif.tbl -g
output_mosaic/Interp_drizzle1/interp_ImageList.txt -c
output_mosaic/Interp_drizzle1/interpolated_covgmap.txt -tu
output_mosaic/Coadd_drizzle1/coadd_Sigma_Tiles_List -to
output_mosaic/Coadd_drizzle1/coadd_tiles.tbl -ti
output_mosaic/Coadd_drizzle1/coadd_tile_bcd.txt
```

Pipeline Module MOSAIC_COADD Version 2.5

Processing time Tue Jan 20 18:11:26 2004

nmimgs 10

Error in allocating memory for Coverage Image for Tile 0

mosaic_coadd Error Message 0x0071: MALLOC_FAILED.

A. This is an indication that you ran out of RAM. The chances are you are trying to make a very big mosaic. First, make sure that this is what you actually want to do. Check the keywords `NAXIS1`, `NAXIS2` in the fiducial image frame file `fif.tbl`. It is possible that a huge fiducial image frame was created because one or more images in the set have bad pointing. Verify that all the images have reasonable pointing (`header_list.tbl`). If you believe that the problem is real, i.e. you are actually trying to make a mosaic as big as it is in `fif.tbl`, then use the tiling capabilities by specifying the tile size for the appropriate module in the namelist. For coaddition (`mosaic_coadd`) and `mosaic_covg` the keywords are:

```
TILEMAX_X = 1000,  
TILEMAX_Y = 1000,
```

For multiframe (`mosaic_outlier`) and dual outlier rejection (`mosaic_dual_outlier`) the keywords are:

```
TILE_XSIZ = 500,  
TILE_YSIZ = 500,
```

Q. I have a list of images covering a big area on the sky, but I only want to make a mosaic covering a small portion of that area. How can I do it?

A. First, create the fiducial image frame based on the full list. In order to do so run *mosaic.pl* with only one step set to 1 in the namelist:

```
run_fiducial_image_frame = 1
```

This will create `FIF.tbl`. Now, set `run_fiducial_image_frame = 0`. Modify `FIF.tbl` to specify the reduced field. Then run *mosaic.pl* with the following setting:

```
run_mosaic_geom = 1
```

Specify all the pertinent input lists in the namelist. For each input list there will be a corresponding reduced list created in the output directory (`OUTPUT_DIR`). The lists will contain only the images that have some overlap with the image frame specified in the modified `FIF.tbl`. E.g. if you have the following setting in your namelist:

```
OUTPUT_DIR = mosaic  
IMAGE_STACK_FILE_NAME = data/MIPS70_ImageList.txt  
DCE_STATUS_MASK_LIST = DmaskList.txt
```

After you run *mosaic.pl* there will be two lists created:

```
mosaic/geom_MIPS70_ImageList.txt  
mosaic/geom_DmaskList.txt
```

Set `run_mosaic_geom = 0`. Now use the reduced lists in the namelist:

```
IMAGE_STACK_FILE_NAME = mosaic/geom_MIPS70_ImageList.txt  
DCE_STATUS_MASK_LIST = mosaic/geom_DmaskList.txt
```

and proceed with mosaicing.

The program uses a namelist parameter `MARGIN`. The parameter gives a margin in terms of the number of pixels around the FIF. Normally, one would want to set to a negative number to avoid including input images that have only marginal overlap with the FIF.

Q. After I have run *overlap.pl* the mosaic doesn't have the patchiness, but it now displays an overall gradient. Is it real or was it introduced by *overlap.pl* ?

A. The chances are it is neither of the above. Most likely the overall gradient in the mosaic image comes from the gradients in the input images, which were not properly flat-fielded. Perl script *flatfield.pl* included in the package should take care of this problem.

Q. I have run *medfilter* to create background subtracted images for *overlap.pl*. It took me forever because I used a big window size. I don't want to do it again for *mosaic.pl* and *apex.pl*. Is there a easy way I can reuse those background subtracted images?

A. Yes, now you can. If a subdirectory name is specified as a full path, it is not treated as a subdirectory `OUTPUT_DIR` and the name is used as specified. So all you have to do is to specify `MEDFILTER_DIR` to point to the location where the background subtracted images created by *overlap.pl* were saved. For example, in the *overlap.nl* you have

```
OUTPUT_DIR = output_overlap
MEDFILTER_DIR = Medfilter
```

You are running *overlap.pl* and *mosaic.pl* in the same directory `/home/NGC0001/`.

Then, this is what you write in the *mosaic.nl*

```
MEDFILTER_DIR = home/NGC0001/output_overlap/Medfilter
```

Point source extraction – apex.pl, apex_1frame.pl

Q. I have made a mosaic using *mosaic.pl*. Do I have to redo it when I run *apex.pl*?

A. No, you don't have to redo mosaicing. Script *apex.pl* can use the tiles created by *mosaic.pl* and saved in the coadd directory (`COADDER_DIR`). If the products of *mosaic.pl* are to be used by *apex.pl*, then the following option should be used in the namelist used by *mosaic.pl*:

```
keep_coadded_tiles = 1
```

Then set `OUTPUT_DIR` to be the same in both the *apex.pl* and *mosaic.pl* namelists. The `run_mosaic_int` and `run_mosaic_coadd` steps can and should be skipped (set to 0). This is especially relevant if outlier rejection is performed as part of mosaicing, since *apex.pl* does not do outlier rejection.

Q. The detection list – `mosaic_detect.tbl` – has no detections.

A. Parameter `Detection_Threshold` is set too high. Try lowering it.

Q. I don't want to do passive deblending. Is there a way to turn it off?

A. The way to avoid doing passive deblending is to remove the *deblend_size* and *deblend_id* columns. To do so, set the following in the namelist:

```
run_detect_select = 1
```

and add the list of columns to copy:

```
select_detect_columns = "srcid,x,y"
```

The perl script *select* will create a new detection table without the *deblend_size* and *deblend_id* columns and passive deblending will not be performed.

Q. The values of chi2/dof's in the extraction table – *extract.tbl* – are way too high.

A. Check the uncertainty images, they could be underestimated.

Q. All the chi2/dof's in the extraction table – *extract.tbl* – are ~ 0.1 or lower.

A. Check the uncertainty images, they could be overestimated.

Q. Can I do point source extraction on the mosaic image?

A. Yes. First you produce a mosaic image with the corresponding uncertainty image. Then you run *apex_1frame.pl* setting `INPUT_IMAGE_FILE = <mosaic_image_file_name>` and `SIGMA_IMAGE_FILE = <mosaic_uncertainty_image_file_name>`. Use the appropriate PRF file.

Q. I only want to do aperture photometry for a given set of sources in the sky. Is there an easy way to do that without using *apex.pl* or *apex_1frame.pl*?

A. Yes. You can run the module *aperture* on the command line with the appropriate setting of the parameters on the command line or in the namelist. See the document **aperture** for details. There you will also find an explanation on how to measure aperture photometry for a single point in the sky in a set of images.

Q. My *apex.pl* run crashed with the following obscure message:

```
Perl system() return code = 6
sourcestimate ran with signal 6
System Exit Code (      sourcestimate): 6
```

What's going on?

A. Chances are you ran out of RAM because of big blends of detections in the detection table. The program *sourcestimate* performs simultaneous fitting of all the detections in the detection table with the same value of the `blendid` column (passive deblending). In

order to bypass this problem you need to clean the detection table by removing huge “blends” of detections. You can do it by inserting the following two lines in the namelist:

```
run_select_detect = 1
```

```
select_detect_conditions = " blend_size < 100"
```

The actual number for the `blend_size` limit may vary with data. In order to find the number applicable for your data you can run *select* directly on the detect table, like so:

```
select from mosaic_detect.tbl for "blend_size > 100"
```

select will print out the number of rows with `blend_size > 100`. If there are none, you can decrease the limit.