

prf_estimate.pl

David Makovoz, 10/15/04

Table of Contents

prf_estimate.pl.....	1
1 Overview	2
2 Input	2
2.1 Input Data.....	2
2.2 Namelist – Configuration file	3
2.3 Quality Control Mask Images	6
2.4 Other Options	7
3 Processing	7
3.1 Background Subtraction.....	7
3.2 Postage Stamp Image Creation.....	8
3.3 PRF Estimation.....	8
3.3.1 NaN Replacement	8
3.3.2 Exact Point Source Flux and Position Estimation	9
3.3.3 Outlier Rejection.....	10
3.3.4 Resampling and Combining.....	11
4 Output	11
5 References	12
6 Appendices.....	12
6.1 PRF Map	12
6.2 Short description of the intermediate and final products	13

1 Overview

The software in this package performs PRF estimation. (See [\stage\ssc-pipe\docs\working_notes\Post_BCD\PRF_vs_PSF.doc](#) for a short description of the difference between PRF (point response function) and PSF (point spread function).) PRF estimation can be performed for a single image (usually a mosaic image) or a set of images (set of bcd's). The coordinates of the point sources in the input images to be used to derive a PRF are given in an input table. Optionally, background subtraction can be performed on the input images. A set of postage stamp images is cut out from each input image centered on each input point source position. The precise flux and point source positions can be optionally estimated by fitting a Gaussian to the postage stamp images. There is an option of rejecting and replacing outlier pixels and outright rejecting "bad" images. The postage stamp images are resampled and shifted using bicubic interpolation and combined into one final PRF image. The software has an option of deriving a set of PRF's mapped into the detector array.

2 Input

2.1 Input Data

The script *prf_estimate.pl* requires input images and a namelist (configuration) file. Table 1 lists the names of the input files for *prf_estimate.pl*. These names can be set in the namelist file, or on the command line; some of them have a default value. Except for the namelist, the names of the input files can be specified using a relative or the absolute path. The command line settings override the namelist settings.

Input File	Default Name	Namelist name	Command line option	Required
Namelist *	prf_estimate.nl	N/A	-n	y
List of input images	-	IMAGE_STACK_FILE_NAME	-I	y*
Input Image	-	INPUT_FILE_NAME	-i	y*
Point source list	-	POINT_SOURCE_LIST	-E	y
FIF file name	-	FIF_FILE_NAME	-F	n
Permanently damaged pixels mask image	-	PMASK_FILE_NAME	-M	n
List of DCE status mask images	-	DCE_STATUS_MASK_LIST	-d	n

List of outlier Rmask images	-	RMASK_LIST	-R	n
------------------------------	---	------------	----	---

Table 1 Input data for *prf_estimate.pl*. *- namelist needs to be in cdf / subdirectory. *Only one out of the two is required. If both a list of input images and an input image are given, the list will be used.

2.2 Namelist – Configuration file

The namelist contains several blocks of various parameter settings, input image names, and running options. Most of the parameter settings for the modules are in the corresponding blocks delineated by “&” following by the capitalized module name in the beginning and “&END” at the end of the block. Several parameters affecting more than one module are set outside of the individual modules’ blocks. Also, the locations of the output final and intermediate products are set in the namelist file.

Table 2 lists the names of the modules, along with their purpose and namelist triggers. To run a module, its trigger should be set to 1.

Module	Namelist trigger	Purpose
<i>medfilter</i>	run_medfilter	Background subtraction of the input images
<i>crop_stack</i>	run_crop_stack	Cut out postage stamp images from the input image around each of the point source on the point source list.
<i>prf_estimate.pl</i>	split_by_array_position	Splits the input image list into separate lists, one list per detector array sector (see 6.1 PRF Map)
<i>prf_estimate</i>	run_prf_estimate	Estimate the PRF

Table 2 Modules, their namelist triggers, and purpose.

The intermediate and final products are written in several subdirectories. The names of the subdirectories can be configured in the namelist file. Table 3 lists the default names of the output subdirectories, the keywords used in the namelist, and all the products written in the subdirectory. OUTPUT_DIR can be specified as a relative or absolute path.

Subdirectory set in Namelist	Default name	Output files
OUTPUT_DIR	./	PRF.fits, PRFVar.fits, copy of the namelist, and all other subdirectories listed below
MEDFILTER_DIR	Medfilter	Background subtracted input images
CROP_STACK_DIR	Crop_Stack	Interpolated images

Table 3 Output directories and the products written there.

The subdirectories are created by *prf_estimate.pl*. For the complete list of products, see Table 6.

Table 4 lists all the parameters, along with their default values, if any, a short description, and the name of the module(s) using this parameter.

The input variables require a space preceding and following the equal sign, i.e. "variable = value". An entry like "variable=value" will not be read and the hard-coded default will be used so you will not notice that your input value is not being used.

Parameter Name	Description	Default	Module
USE_REFINED_POINTING	Switch to use refined pointing keywords (int)	0	<i>crop_stack</i>
PRFMAP_FILE_NAME	The filename of the PRF Map (char)	-	<i>prf_estimate.pl</i>
PRF_file_name	The filename of the output PRF fits image(char)	PRF.fits	<i>prf_estimate</i>
PRF_Sigma_file_name	The filename of the fits image of the uncertainty of the output PRF (char)	PRFSigma.fits	<i>prf_estimate</i>
PRF_Coverage_file_name	The filename of the fits image of the coverage of the output PRF (char)	-	<i>prf_estimate</i>
Star_file_name	The filename of the fits image of a point source (char)	-	<i>prf_estimate</i>
StatusList_file_name	The file name of the status file, that lists all the input images and whether or not they were used to produce the PRF along with the reason they were rejected(char)	-	<i>prf_estimate</i>
AnnealStatus_file_name	The file name of a table with some diagnostic information about simulated anneal fitting(char)	-	<i>prf_estimate</i>
Window_X Window_Y	x-size and y-size in pixels of the sliding window placed around each pixel to estimate the median (int)	-	<i>medfilter</i>
N_Outliers_Per_Window	Number of pixels excluded from the sliding window described above (int)	0	<i>medfilter</i>

PostStamp_Xsize PostStamp_Ysize	The size in pixels of the postage stamp image (int)	-	<i>crop_stack</i>
X_Column_Name	The column name of the column with the first coordinate (char)	RA	<i>crop_stack</i>
Y_Column_Name	The column name of the column with the second coordinate (char)	Dec	<i>crop_stack</i>
TransformationType	The coordinate transformation type (char). The choices are 'SkyToPlane', 'PlaneToPlane', 'NoTransform'	-	<i>crop_stack</i>
MinNumber_Replacement_Pixels	Minimum number of neighboring pixels to replace a NaN pixel with their median (int)	8	<i>prf_estimate</i>
Reject_NaNs	The maximum number of NaN pixels in an image before the image get rejected(int)	0	<i>prf_estimate</i>
Outlier_Threshold	The number of sigma's above or below the trimmed mean to reject outliers.	3	<i>prf_estimate</i>
Min_Outlier_Overlap	Minimum overlap area in pixels between the interpolated pixels detected as outliers and the original pixels of the input postage stamp images (float)	0.5	<i>prf_estimate</i>
Max_Ratio_Outliers*	Maximum ratio of the number of outliers in an input image to the input image size before this image is excluded completely from the PRF estimation (float)	0.1	<i>prf_estimate</i>
Max_Number_Outliers*	Maximum number of outliers in an input image before this image is excluded completely from the PRF estimation (int)	0	<i>prf_estimate</i>
Flux_Column_Name	The column name of the column with the flux (char)	'flux'	<i>prf_estimate</i>
PRF_ResampleX_Factor PRF_ResampleY_Factor	The ratio of the input image pixel size to the sampling distance in x- and y- directions	-	<i>prf_estimate</i>

	(int)		
Zero_Edge	The size in pixels of the margin of the RPF to be set to 0(int)	0	<i>prf_estimate</i>
Fit_Seed	The seed for random number generation to be used by simulated anneal fitting. If set to 0 the current time is used. If negative, fitting is not performed(int)	0	<i>prf_estimate</i>
StatusList_File_Name	The file name of the status file, that lists all the input images and whether or not they were used to produce the PRF along with the reason they were rejected(char)	-	<i>prf_estimate</i>
Fit_X, Fit_Y	Fitting area size in pixels (int)	5	<i>prf_estimate</i>
Center_Box_X, Center_Box_Y		7	<i>prf_estimate</i>

Table 4 The processing parameters for *prf_estimate.pl*. The shaded fields are for the parameters set outside of the individual module blocks. *Both conditions have to be met in order for an image to be excluded.

2.3 Quality Control Mask Images

Quality control mask images can be used in the processing, specifically in producing the background subtracted images. Three kinds of mask images can be used: permanently damaged pixels masks (Pmask), DCE status masks (Dmask), and outlier masks (Rmask). Normally there will be a single Pmask for a set of input images.

Each bit of the pixel value in a mask image corresponds to a particular condition. A fatal bit pattern is a short integer that has the bits of interest set. Each pixel in a mask image is matched against the fatal bit patten. If any of the bits specified by the fatal bit pattern is set in the value of a pixel in a mask image, then in the corresponding image the corresponding pixel is considered unusable.

If the name of the mask image or list of mask images is not set the scripts will proceed without using them. If they are set then the corresponding fatal bit pattern should be specified in the namelist.

Mask Name	Fatal Bit Pattern Name
Pmask	PMask_Fatal_BitPattern
Dmask	DCE_Status_Mask_Fatal_BitPattern
Rmask	RMask_Fatal_BitPattern

Table 5: Setting of fatal bit patterns for various mask images in the namelist file.

If one of the bits set in a fatal bit pattern is also set in a pixel of the corresponding mask, then the value of the corresponding pixel in the background subtracted image is set to NaN and is ignored for further processing.

2.4 Other Options

1. Switch `NICE`. If `NICE = 1` all the modules called by the script with “nice 19.” The default is 0.
2. Switch `save_namelist`. The namelist used in the current run is always copied to the output directory. By default the name of the namelist is not changed. By setting `save_namelist = 1` in the namelist the namelist copied to output directory will be given a unique name, which is created by appending the namelist name to the time of execution. For example, if you ran “`prf_estimate.pl -n myname.nl`” at 12:32:53, then the namelist will be copied to the output directory as `12h32m53s_myname.nl`. The default is 0, in which case the file is copied as `myname.nl`.
3. Switch `delete_intermediate_files`. If `delete_intermediate_files = 1` is set in the namelist the products of all the modules run this time will be deleted except for the last module. The default is 0.

3 Processing

3.1 Background Subtraction

Module *medfilter* performs background estimation in the input images and outputs background subtracted images.

The program computes an asymmetrically skewed median for each pixel in the input image using a rectangular window of `Window_X` by `Window_Y` size. It is achieved by omitting `N_Outliers_Per_Window` highest pixels from each median window. If `N_Outliers_Per_Window` is set to 0 the program calculates the regular median.

There is a minimum required number of good (not-NAN and not marked by any mask) pixels per median window `Min_Good_Pixels_In_Window`. If the number of good pixels is below this threshold then the corresponding pixel in the output image is marked as a “missing” pixel. When the median calculation is finished, the values of the marked pixels are interpolated from the neighboring pixels for which the median has been found. In order to do so the program scans around the pixel in question and accumulates values of good pixels. When the number of accumulated values reaches or exceeds the minimum number given by the input parameter `Min_GoodNeighbors_Number` the program finds the average and stores this value as the median for the pixel in question.

If the number of “missing” pixels exceeds the maximum number given by the parameter `Max_Bad_Pixels_OutputImage`, the program aborts, printing the appropriate error message.

The input image pixels masked by the input R-,DCE_Status-, or P- masks are set to NaN in the output background subtracted images.

3.2 Postage Stamp Image Creation

Module `crop_stack` cuts out postage stamp images from the input images around each point source from the point source list (`POINT_SOURCE_LIST`). If a single image name (`INPUT_FILE_NAME`) is specified the postage stamps are cut out from that image. If a list of images (`IMAGE_STACK_FILE_NAME`) is specified, then the postage stamp images are cut out from each image listed therein. The postage stamp images are saved in the output directory specified by two namelist variables:

`OUTPUT_DIR/CROP_STACK_DIR`. The naming convention for these images is *img_I_J.fits*. Here *I* is the point source number in the point source list, *J* is the image number in the list of images. Their size is specified in the namelist using two parameters `PostStamp_Xsize` and `PostStamp_Ysize`. The positions of the center of the postage stamps in the input images are given in the input point source list (`POINT_SOURCE_LIST`). The position can be given in sky coordinates or in pixel coordinates. The appropriate coordinate transformation type should be specified in the namelist (`TransformationType`). If the positions are given in sky coordinates then the transformation type should be set to `'SkyToPlane'`. If the positions are given in pixel coordinates in some image frame (e.g. mosaic image), then the transformation type should be set to `'PlaneToPlane'`. In this case a file describing the image frame should also be given (`FIF_FILE_NAME`). (FIF stands for Fiducial Image Frame.)

3.3 PRF Estimation

Module `prf_estimate` computes a PRF image(s) based on the input images of a single isolated point source, not necessarily the same source, just a single source. An attempt is made to automate the process of weeding out bad images from the process of PRF estimation using several outlier rejection mechanisms and using data fitting to determine input point sources' flux and positions. However, it is up to the users to verify that these efforts were ultimately successful, especially if they suspect that the PRF produced by the program has some problems.

The software has an option of creating a single PRF image. Alternatively a set PRF images can be estimated for the sectors of the detector array. The mapping of the detector array into sectors is specified in the file called a PRF map (see PRF Map). The namelist list parameter `PRFMAP_FILE_NAME` is used to specify the file name of the PRF map. If a PRF Map is used, the list of input Postage Stamp images should be separated into several lists, one list per detector array sector. In order to do so the namelist switch `split_by_array_position` should be turned on. After that module `prf_estimate` is run consecutively on each partial list and a PRF image is created and saved for each detector array sector.

There are four steps in creating a PRF: NaN replacement, exact point source position and flux estimation, outlier rejection, and coadding.

3.3.1 NaN Replacement

Optionally the program replaces each NaN in the input image with the median of a number of the neighboring pixels. If the parameter `Reject_NaNs` in the namelist is set to 0, which is the default, the replacement is performed regardless of the number of

NaN's in the image. If `Reject_NaNs > 0`, then bad pixels are replaced only if their number is less than or equal to `Reject_NaNs`. Otherwise the image is excluded from the list of images to be used for PRF estimation. Another reason for rejecting an image is if a NaN pixel is found in the center of the input image. The size of the center is specified with the namelist parameters `Center_Box_X`, `Center_Box_Y` both defaulting to 7. The minimum number of pixels to be used for computing the median is given by the namelist parameter `MinNumber_Replacement_Pixels`.

3.3.2 Exact Point Source Flux and Position Estimation

For both outlier rejection and coadding each input postage stamp image shifted to the common grid, in which the peak of the PRF – the point source position - coincides with the center of the central pixel. By default the position of the point sources in the input images are computed as flux weighted centroids. Each postage stamp image is normalized by its flux. The flux can be given in the input point source list. If it is, then the column name of the appropriate column should be 'flux' or can be specified in the namelist by the parameter `Flux_Column_Name`. If the flux is not given in the input table, then it is found by summing all the pixel values in the input image.

The flux and the exact position of the point source in each Postage Stamp image are estimated by fitting the data with a Gaussian. The following quantity is minimized:

$$\mathbf{c}^2 = \sum_{i \in W} (s(i) - G(i))^2$$

$$G(i) = \frac{f}{2\mathbf{p}} \cdot \exp\left(-\frac{(x_i - x_c)^2 + (y_i - y_c)^2}{2\mathbf{p}\mathbf{s}^2}\right)$$

with respect to the flux f , width \mathbf{s} , and position (x_c, y_c) of the point source in the Postage Stamp image. Here $s(i)$ is the pixel value of the input image for pixel i , W is the area in center of the image. The size of the area is specified with the namelist parameters `Fit_X`, `Fit_Y` both defaulting to 5 pixels. Simulated annealing method is used to minimize \mathbf{c}^2 .

3.3.3 Outlier Rejection

Two kinds of outlier rejection are performed. The first kind is when the whole image is rejected based on some criteria. The second kind is when only some pixels in an image are detected as outliers and then replaced with some combination of the neighboring “good” pixels.

Image rejection is done based on several different criteria. Two of the criteria are listed above. They are: having too many NaN’s in the input image, or having a NaN pixel in the central box. Two criteria are based on the results of Gaussian point source fitting (see previous section), specifically using the values of \mathbf{s} and \mathbf{c}^2 for each image. The third criterion is the number of outlier pixels detected in the image. The fourth criterion is having an outlier pixel in the central box of the image. The aggressiveness of outlier rejection is controlled by the namelist parameter `Outlier_Threshold`.

After Gaussian fitting is completed for all input images the sets of \mathbf{s} ’s and \mathbf{c}^2 ’s are analyzed. Image j is declared an outlier, if any of the three conditions below are met:

$$\mathbf{s}_j > \mathbf{s}_{median} + \text{Outlier_Threshold} \cdot \mathbf{s}_s$$

$$\mathbf{s}_j < \mathbf{s}_{median} - \text{Outlier_Threshold} \cdot \mathbf{s}_s$$

$$\mathbf{c}^2_j > \text{Outlier_Threshold} \cdot \mathbf{c}^2_{median}$$

Here \mathbf{s}_{median} and \mathbf{c}^2_{median} are the median values of the corresponding quantity and \mathbf{s}_s is the standard deviation of the Gaussian widths for the set of the input images.

Next step is detecting and rejecting pixels outliers. For the outlier rejection step no resampling is performed. The input images are shifted to the common grid using the bicubic interpolation. The shifted images are stacked up and outliers are found. Namelist parameter `Outlier_Threshold` specifies the number of trimmed sigmas below and above the trimmed mean to be used to determine the outliers. If a particular image has too many outliers it is completely rejected from further processing. How many is too many is determined by two parameters specified in the namelist: `Max_Ratio_Outliers` and `Max_Number_Outliers`. If the fraction of the outliers in an input image is greater than `Max_Ratio_Outliers` and the total number of the outliers is greater than `Max_Number_Outliers` then the image is rejected. An image is also rejected if the outlier pixel is found in the center of the image, the center being defined the same way as it is for the NaN pixels: it is specified with the namelist parameters `Center_Box_X`, `Center_Box_Y` both defaulting to 7. If an image has a number of outliers but is not rejected, then the values of the outlier pixels is replaced with the median of the neighboring pixel values in exactly the same way the NaN’s are replaced in the input images. There is an additional step of re-projection of the shifted images back on the input images, since the outlier detection is performed on the shifted images. A pixel in an outlier image is considered an outlier if the fractional area overlap of the pixel with the outlier pixels from the shifted image is greater than `Min_Outlier_Overlap`.

3.3.4 Resampling and Combining

For coadding the input images are resampled and shifted to the common grid using the bicubic interpolation. The ratio of the pixel sizes to the sampling distance is given by the namelist parameters `PRF_ResampleX_Factor` and `PRF_ResampleY_Factor`. The resampled and shifted images are stacked up and a simple mean and standard deviation are found for each pixel position.

4 Output

If `PRFMAP_FILE_NAME` is specified in the namelist, then the names of the output files are read from the PRF map (see 6.1 PRF Map section), in which case `PRF_file_name` and `PRF_Sigma_file_name` are ignored if given. Otherwise they can be used to specify the names of the output images with the default names `PRF.fits` and `PRFSigma.fits`. A coverage map is output if `PRF_Coverage_file_name` if given in the namelist.

Optionally an image of a point source is written in a fits file with the name specified by the namelist parameter `Star_file_name`. The image is created using the computed PRF and placing the point source at the center of the central pixel of the image.

`StatusList_File_Name` is the name of the file that lists all the input images and whether or not they were used for PRF estimation, and if not, the reason for rejection is listed. Here are some excerpts from such a file:

```
image #0 CropStack_0.1_2/img_1_1.fits has an outlier in the central box
image #1 CropStack_0.1_2/img_1_2.fits was used for prf estimation
image #2 CropStack_0.1_2/img_1_3.fits has too many NaNs
image #3 CropStack_0.1_2/img_1_4.fits was used for prf estimation
image #4 CropStack_0.1_2/img_1_5.fits couldn't replace NaN in pixel 1
image #339 CropStack_0.1_2/img_9_2.fits has a chi2 outlier
image #390 CropStack_0.1_2/img_12_1.fits has a NaN in the central box
image #1259 CropStack_0.1_2/img_22_678.fits has too many outliers
image #1260 CropStack_0.1_2/img_22_679.fits has too many NaNs
image #1636 CropStack_0.1_2/img_27_14.fits has a sigma outlier
```

`sadfAnnealStatus_file_name` is a table with some diagnostic information about simulated anneal fitting. Here is an example of such a table:

```
\char comment = Output from PRF_ESTIMATE, version 2.00
\char Date-Time = Tue May 11 09:03:16 2004
\char comment = dlimage library version 2.00
\char Input_Image_List = mips3865856_list.txt
\int MinNumber_Replacement_Pixels = 8
\int Reject_NaNs = 0
\char PRF_Filename = PRF_new.fits
\char PRFSigma_Filename = PRFSigma_new.fits
\char PRFCoverage_Filename = PRFCov_new.fits
\int PRF_ResampleX_Factor = 4
\int PRF_ResampleY_Factor = 4
\float Alpha = -0.5
\float Outlier_Threshold = 3
\int Max_Number_Outliers = 0
\float Max_Ratio_Outliers = 0.1
\float Min_Outlier_Overlap = 0.5
\int Zero_Edge = 0
```

```

\char FitStat_File_Name = fit_stat.txt
\int Fit_Seed = 1
|          image_id|          sigma^2|          chi^2|
|i              |r              |r              |
      1          4.362e+00      1.309e-03
      2          3.874e+00      7.051e-04
      3          3.803e+00      5.988e-04
      4          3.882e+00      4.386e-04
      5          3.742e+00      4.074e-04
      6          3.871e+00      1.021e-03
      7          4.017e+00      6.598e-04
      8          3.972e+00      6.530e-04
      9          4.009e+00      1.012e-03
     10          3.794e+00      6.453e-04
     11          4.113e+00      4.050e-04
     12          3.915e+00      5.354e-04
     13          3.960e+00      4.290e-04

```

5 References

All of the documents referenced in this document can be obtained from the Spitzer Science Center website, <http://spitzer.caltech.edu/SSC/>. They include:

1. PRF_vs_PSF.

6 Appendices

6.1 PRF Map

A PRF map divides the detector array into a number of sectors, in order to have a separate PRF image for each sector. The keyword `Number_PRF` gives the number of such sectors and, consequently, different PRF's that should be used for an image. The names of the files containing the PRF's are given by the keywords `PRF_Filename_#`, where `"#"` runs from 1 to `Number_PRF`. The keyword `haveSigma` is used to indicate whether PRF uncertainty images should be made. If they are requested, `haveSigma = 1`, `HaveSigma = 0` otherwise. The names of the images with the Sigmas are given by the keywords `PRFSigma_Filename_#`. The keywords `ImageX` and `ImageY` give the sizes of the images for which the PRF's are to be used.

Here is a sample PRF Map file.

```

\char comment = PRF Map

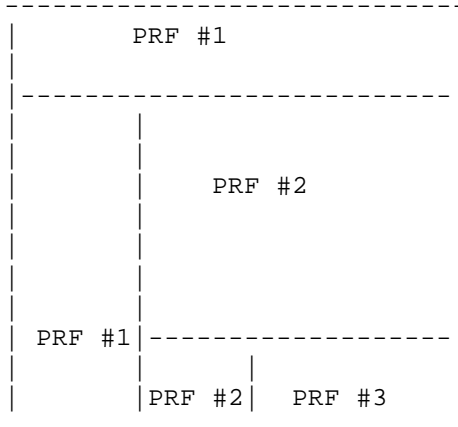
```

```

\int Number_PRF = 5
\int HaveSigma = 1
\int ImageX = 256
\int ImageY = 256
\int PRF_Filename_1 = /ssc/pipe/davidm/IRAC.2.PRF.1.fits
\int PRF_Filename_2 = /ssc/pipe/davidm/IRAC.2.PRF.2.fits
\int PRF_Filename_3 = /ssc/pipe/davidm/IRAC.2.PRF.3.fits
\int PRF_Filename_4 = /ssc/pipe/davidm/IRAC.2.PRF.4.fits
\int PRF_Filename_5 = /ssc/pipe/davidm/IRAC.2.PRF.5.fits
\int PRFSigma_Filename_1 = /ssc/pipe/davidm/IRAC.2.PRF.Sigma.1.fits
\int PRFSigma_Filename_2 = /ssc/pipe/davidm/IRAC.2.PRF.Sigma.2.fits
\int PRFSigma_Filename_3 = /ssc/pipe/davidm/IRAC.2.PRF.Sigma.3.fits
\int PRFSigma_Filename_4 = /ssc/pipe/davidm/IRAC.2.PRF.Sigma.4.fits
\int PRFSigma_Filename_5 = /ssc/pipe/davidm/IRAC.2.PRF.Sigma.5.fits
|PRFNum|  NAXIS1| NAXIS2|  PRFPos1|  PRFPos2|
|i      |i      |i      |i      |i      |
|-----|-----|-----|-----|-----|
| 1     | 100   | 200   | 1     | 1     |
| 2     | 256   | 56    | 1     | 201   |
| 3     | 50    | 50    | 101   | 1     |
| 4     | 156   | 150   | 101   | 51    |
| 5     | 106   | 50    | 151   | 1     |

```

The geometry for the above table file:



6.2 Short description of the intermediate and final products

The following table lists the names of the intermediate products of *prf_estimate.pl* based on the assumption that the input images have names `<input_image_name>.fits`.

Description	Naming convention	Default Location
PRF fits file, PRF uncertainty fits file, PRF coverage fits file, an image of a point source fits file, status files.	PRF.fits, PRFSigma.fits, PRFCov.fits, <Star_file_name>, <StatusList_file_name>, <AnnealStatus_file_name>	./

The list of the background subtracted images	MedSubtractList.txt	Medfilter/
Background subtracted images	<input_image_name>_minback.fits	Medfilter/
The list of the postage stamp images	CropImageList.txt	Crop_Stack/
Postage stamp images	img_I_J.fits	Crop_Stack/

Table 6. A list of the intermediate and final products of *prf_estimate.pl*.