



imclean

IRAC artifact correction routines

Joseph Hora
Center for Astrophysics | Harvard & Smithsonian
2020/09/24

Table of Contents

1. Introduction.....	1
2. Setup and initialization of imclean	2
2.1. Files and directories.....	2
2.2. iraf configuration	2
2.3. Editing the files.....	3
2.4. Starting up the program	3
2.4.1. imclean command line parameters.....	4
2.4.2. imclean script operation	5
3. Commands in imclean	6
3.1. Banding correction (d or D command).....	9
3.2. Column Banding correction (m or M command)	11
3.3. Warm mission column pulldown corrector (w command).....	11
3.4. Column pulldown – section of column (v and V commands).....	12
3.5. Row and column “pulldown” correction commands (c, l, r, k).....	13
3.6. Masking regions (a command)	13
3.7. Muxbleed correction (b or B command)	13
3.8. Striping correction (s and S commands).....	15
3.9. Pixel interpolation (p or P command).....	16
3.10. Overwrite file with original BCD file (o command)	17
4. spotclean – an imclean variant.....	19
5. Change log (after 2019/07/23).....	19

1. Introduction

The `imclean.cl` routine is an iraf script that allows the user to visually examine a set of IRAC BCDs or cBCDs to find frames where there are residual artifacts¹, and to identify the artifacts interactively in each image and correct them. The `imclean` program is a simple iraf script that accepts cursor position input from the `SAOimageDS9` program² and calls other stand-alone c programs or iraf scripts that read in the image, perform the corrections, and write the image back to the original file name. In most cases the image is then redisplayed so the user can evaluate the effectiveness of the correction, and if necessary cancel the operation and return to the original image. The user controls how these programs operate by specifying locations in the image to perform the corrections.

The stand-alone programs that actually perform the corrections can be run from the linux command line outside of the `imclean` script. See the `imclean.cl` program for how to call the programs with the necessary command line parameters. Or run the programs with no parameters and they will display a list of the necessary inputs and options. For most of the corrections, it is necessary to specify the image section to perform the correction properly. If one had an automated way to examine the images and call the correction programs, using the programs from the command line might be an efficient way to perform the corrections. However, in general, the IRAC images in regions with extended ISM emission can be complicated, and it is difficult to automatically find and correct many types of artifacts that appear in the IRAC images. Therefore, the `imclean` routine was written to make it easy to do the artifact corrections.

The program requires iraf, some version of `SAOImageDS9` that iraf is configured to send images to for display, and the properly compiled c programs for your system. The c programs are in the “src” subdirectory in the distribution. The data are assumed to be in a format similar to the original BCD, i.e., 256x256 pixel images, ending in `*bcd.fits`, with mask and other files located in the same directory. It also works well on data that have been ingested with `IRACproc` (Schuster et al. 2006, SPIE, 6270, 627020; <https://www.cfa.harvard.edu/~mmarengo/IRACproc/>). One could also work with the new “CBCD” data, there is another configuration parameter in the `imclean` script that can be changed to work with those filenames (see Section 2.3 below).

There are a few necessary tasks to perform before running `imclean` to configure iraf and the scripts for your environment, these are described in the following section.

¹ see Hora et al. 2004, Proc. SPIE, 5487, 77

² <http://ds9.si.edu/site/Home.html>

2. Setup and initialization of imclean

2.1. Files and directories

The following is a suggested setup configuration for the software. It is possible to set it up differently, but this is one way to make it work on your system.

In your iraf home directory, make a directory called `imclean_dist`, and unzip the imclean distribution file in that directory. This will create the following files and directories:

```
imclean.cl
initclean.cl
spotclean.cl
normmedian.cl
imclean_docs.pdf
README
/bin
/src
```

The last two are directories with the binaries and source for the c programs. The binaries in the distribution file were compiled on a CentOS linux 64-bit Intel processor system. If the binaries cannot be run on your system, you will have to recompile the programs using the source code in `/src` and copy the executables to the `/bin` directory. The programs can be recompiled as follows, for example:

```
cc rowcorrect.c fitsio.c swapbytes.c -lm -o rowcorrect
```

for the program “rowcorrect”. Along with the target program `rowcorrect`, the files `fitsio.c` and `swapbytes.c` are compiled which include necessary subroutines for the main program. All of the other programs are compiled in a similar manner, except for the program `chanslice`, which compiles with a special version of `fitsio` called `fitsio_cs.c`:

```
cc chanslice.c fitsio_cs.c swapbytes.c -lm -o chanslice
```

There is also a script called `compile_imclean.sh` in the `src` directory which runs all of the compiler instructions to generate the executables and then transfers the executable files to the `../bin` directory (see Section 2.3). This script can be used instead of compiling the programs one-by-one. You might have to edit this script for the specific setup of your system.

2.2. iraf configuration

In your `login.cl` file for iraf, make sure the parameter “stdimage” is set to `imt4096` or larger. Set these other parameters as well:

```
imtype = "fits"
imdir = ""
```

and be sure to uncomment the lines (remove the “#” at the beginning of the lines!).

2.3. Editing the files

There are two customizations that must be done to the iraf scripts so that they can find the executable files for the various image correction functions. In the script “`initclean.cl`”, change the “`/home/jhora/iraf`” path to the directory where you have put the `imclean_dist` directory tree. Then move the `initclean.cl` script to your iraf home directory, so you can easily run it after logging in to iraf (see Section 2.4).

Then in `imclean.cl` and `spotclean.cl`, change the line

```
bindir="/home/jhora/iraf/imclean_dist/bin/"
```

to refer to the location of the `imclean` “bin” directory on your system. If you have followed the instructions in the previous section, this would point to a subdirectory of your iraf home directory, ending in “`imclean_dist/bin/`”.

There are a couple other variables defined in the `imclean` script that also might need to be changed. The variable “`masksuffix`” is a string that specifies how the mask files are named. This is needed when the program is trying to set mask bits when the user has entered the “a” command. The suffix is set by default to “`bimsk`” and “.fits” is added to the end. If your files are named “`bdmsk`” or some other specification, change this variable to match your mask file names.

Another parameter is “`suffixlen`”, which is by default set to 8. This tells the program how many characters are at the end of the image file name, after the root that is common to the BCD, mask, and noise images. For example, if the BCD file name is

```
SPITZER_I1_32700416_0028_0000_2_bcd.fits
```

then the `suffixlen=8` (for the characters “`_bcd.fits`”). If instead the file ends in “`_cbcd.fits`”, then the `suffixlen=9`.

2.4. Starting up the program

To run the `imclean` program, start `iraf` on your system like you normally do. Then in `iraf` do the following command (the “`cl>`” at the start of the line is meant to show the command prompt, do not type that):

```
cl> cl < initclean.cl
```

This runs the command to define the scripts (you could also put the contents of `initclean.cl` in your `login.cl` file if you prefer so that it runs every time that you start `iraf`). The `initclean.cl` script also loads the package `noao.imred.ccdred.ccdtest` that is necessary for one of the functions in `imclean`, and defines the `normmedian` command.

Then start SAOImageDS9 in the way you normally do to display images from within iraf. This could be for example:

```
cl> !ds9 &
```

The images are displayed by imclean using the iraf “display” command. This remaps the values in the image to the display, depending on what scaling scheme has been selected. The default is a linear scaling with min and max values that you have entered on the command line (see below). This can be altered by changing the lines in imclean that run the display command to use some alternate scaling.

Change your current directory to the location of the files you want to work on. Then you can start the imclean program to start working on the files. It is most convenient if you make a file containing a list of files you want imclean to work on. If you are using IRACproc, the data directory would be something like “/X/object/reduce”, and the “/X/object/Lists” directory has the lists of the files you will work on, with the name something like `bcd_I1.lis`. Or you can make your own file list, for example:

```
ls I1*bcd.fits > filelist
```

So for example, here is a command to start up imclean from the “reduce” directory:

```
cl> imclean @filelist 1 -1 10 1 /data/PID1000/AORs/
```

2.4.1. imclean command line parameters

There are five command line parameters. The first is the list of the files to work on. The file list could be a wildcard expression, such as “I1*bcd.fits”, or refer to a file that contains a list of files, one per line. If referring to a list of files, one uses the usual iraf notation of the “@” symbol in front of the name of the file containing the list. One could also run the program on a single file.

The second parameter on the imclean command line is the set of images within that list to work on. Your list might contain hundreds of files, but imclean can only display and work on sets of up to 100 at a time. So this second parameter tells it which group of 100 to work on. For example, a “1” will select files 1-100 in the list. A “5” would select files 401-500. So to process all of the files in your list, you would just run imclean multiple times, each time incrementing this second parameter.

The third and fourth parameters are the range to scale the data to for the display, the “z1” and “z2” parameters for the “display” command in iraf. Usually you are correcting artifacts that are at a fairly low level compared to the brightest objects in the image, so it is best to select a small range of values near the lowest level in order to enhance the visibility of the artifacts and the quality of the correction. For Channels 1 and 2, a range of -1 to 10 works well in many cases. For Channels 3 and 4, a range of -1 to 100 is good.

The fifth command line parameter is the IRAC channel number (1, 2, 3, or 4). This is used only for the warm mission column pull-down corrector, which differs from the cryo mission correction and uses different correction parameters for Channels 1 and 2.

The last `imclean` command line parameter is the directory where the original pipeline files from the AOR exist, in the format that they are unpacked from the SSC. In the example above, this directory is `"/data/PID1000/AORs"`. In this directory would be the AOR directories, which start with the letter "r" and the rest is the AOR key, for example `"r32700416"`. The `imclean` command "o" will copy the original BCD file from this directory into your current working directory and overwrite the file you are working on (assuming it has the same file name). See the discussion below in Section 3.10 about use of this command to correct cBCD files. This fifth parameter is not strictly necessary if you do not use the "o" function, so one could put a dummy value in this parameter if desired, or if the original BCD files are not available.

2.4.2. `imclean` script operation

After entering this command line, the program reads in all of the files and copies them to a master mosaic which is a single file containing all of the images side-by-side (see Figure 1). The mosaic is then displayed to SAOImageDS9 and the cursor should automatically switch over to the image window for the user to enter cursor commands. At this point, the user could adjust the grayscale with the mouse (within the pre-defined `z1` and `z2`) and zoom and pan the image. This display allows the user to examine all the files at a glance and see the ones that need correction. To select an image to work on, the user moves the cursor to point to the image to be corrected and presses any key. The program then displays the selected single BCD image to work on.

Once a single image is displayed, the subsequent commands require two keys to operate. As many corrections as necessary can be done to an image in succession, each one being applied after the previous one has been completed. When the image has been corrected to the user's satisfaction, one can either move on to the next image, go back to the 100 image display to select a new image, or quit the program. See the sections below for more description of the operation of the program and the individual corrections that can be applied.

The script by default creates a text log file called `"imclean_commands.log"` which records each operation performed on the images. The file is saved in the current directory. Multiple runs are stored in the same file, appended to the end with a timestamp telling when the script was started and the command line parameters used. The log file keeping can be turned off by changing the value of `"keeplog"` to zero in the `imclean` script.

3. Commands in imclean

Below are the commands that one can use when working on an individual image with imclean. These commands are entered as single keystrokes (without pressing the “enter” key) when the cursor is over the image displayed in SAOImageDS9:

```
c - column pulldown correction, whole column
l - column pulldown correction, specified area
r - row pulldown correction, whole row
k - row pulldown correction, specified area
d - row banding correction, whole row, only those > 0.5 sigma from median
D - row banding correction, whole row, force all columns to median
m - column banding correction, whole column, only those > 0.5 sigma from median
M - column banding correction, whole column, force all columns to median
v - banding correction for columns in specified region, only those > 0.5 sigma
  from median
V - banding correction for columns in specified region, force all columns to median
w - warm mission pulldown corrector
a - mask the pixels in the rectangular region (no display refresh)
b - muxbleed correction, row following position
B - muxbleed correction, fix all 4 amplifiers
s - Stripe cleaning
S - Stripe cleaning, selected rows
p - fill selected pixels with interpolated values determined in same row outside
  of selected region (no display refresh)
P - same as "p", except refresh display

n - done editing, go back to mosaic
N - done editing, go to next file in list
u - undo all edits on this image
o - copy original bcd from directory specified on command line, overwrite
  current file
q - quit imclean
```

The top set of commands control the artifact correction. The bottom five commands are for program control: switching between images, undoing the commands, or quitting the program. When editing a single image, one enters the command twice to perform an action. For the first set of commands, the position of the cursor during those two entries defines the region that the artifact routine uses to determine the correction. For some commands, the correction is applied to entire rows, columns, or the whole image; for other commands, the correction is applied to a limited region specified by the cursor locations. For the last set of five commands, the location of the cursor has no effect, but the commands still need two entries. When in the 10x10 image display screen, a single key performs the action of selecting an image, or quitting the program.

The two cursor positions in general define a rectangular image subregion for the task to perform the correction. The locations chosen are the opposite corners of that rectangle. In the examples shown below, the first key specifies the lower left corner and the second key the upper right corner. However, the routines will check the coordinates and reorder them if necessary, so one could select the upper left and lower right corners instead.

Occasionally, you might hit the wrong first key or choose the wrong position. In order to not execute a command, for the second key you can enter a character that is not a valid command, such

as “z”. The program will ignore the input and refresh the display, and you can then enter the valid command or position that you want.

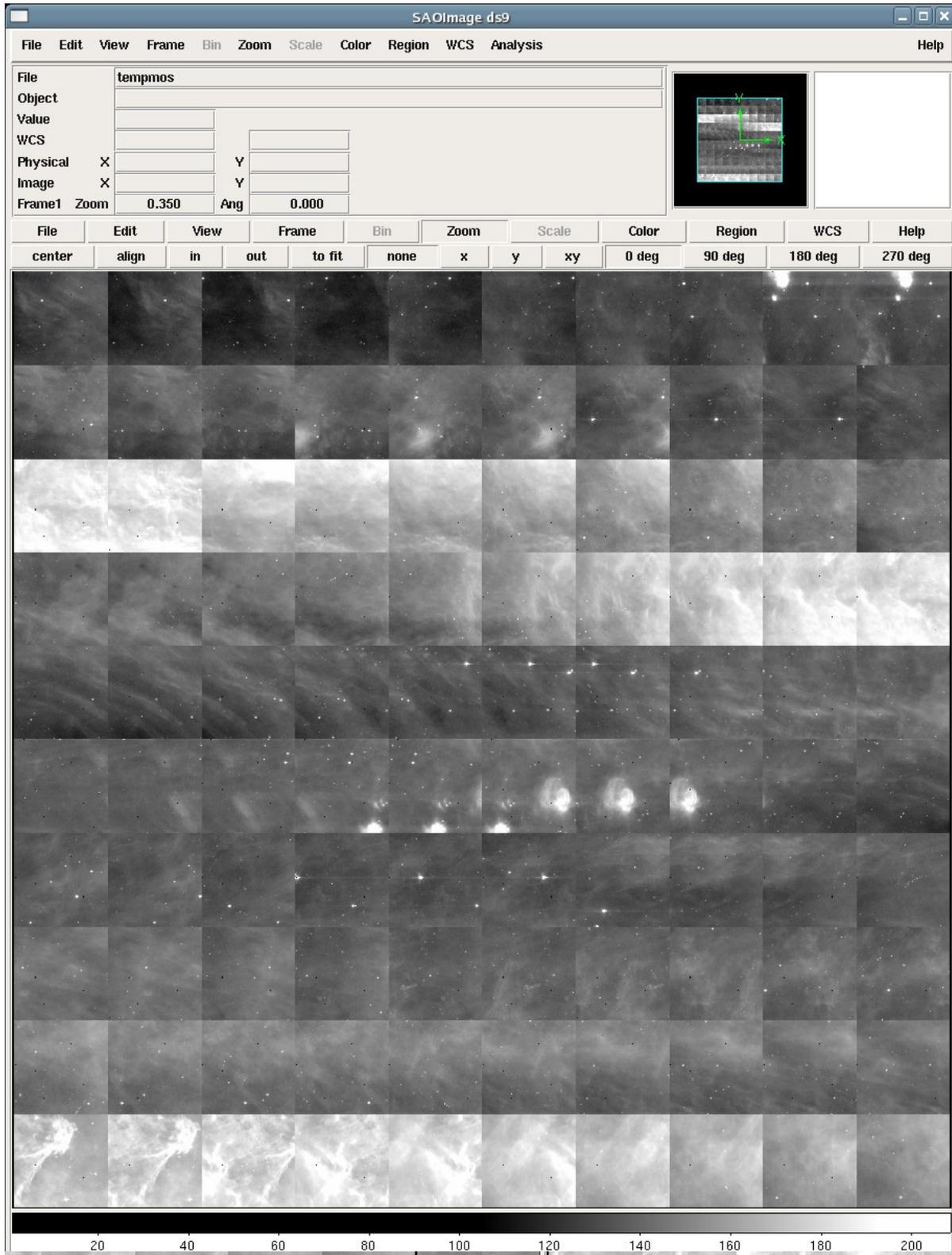


Figure 1. Set of 100 images displayed, ready to choose the image to correct.

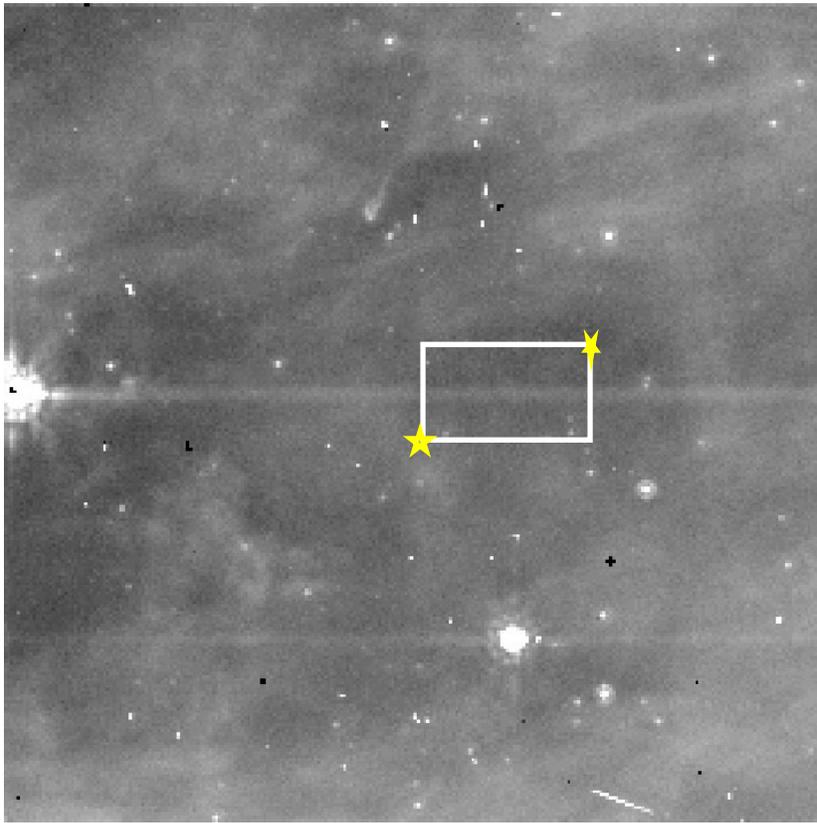


Figure 2. First image to perform banding correction. The stars and white rectangle do not appear in imclean, but have been added for illustration. The yellow stars indicate the two positions that were selected with the cursor and pressing the “D” key. A region of fairly low and uniform background was selected for the correction calculation.

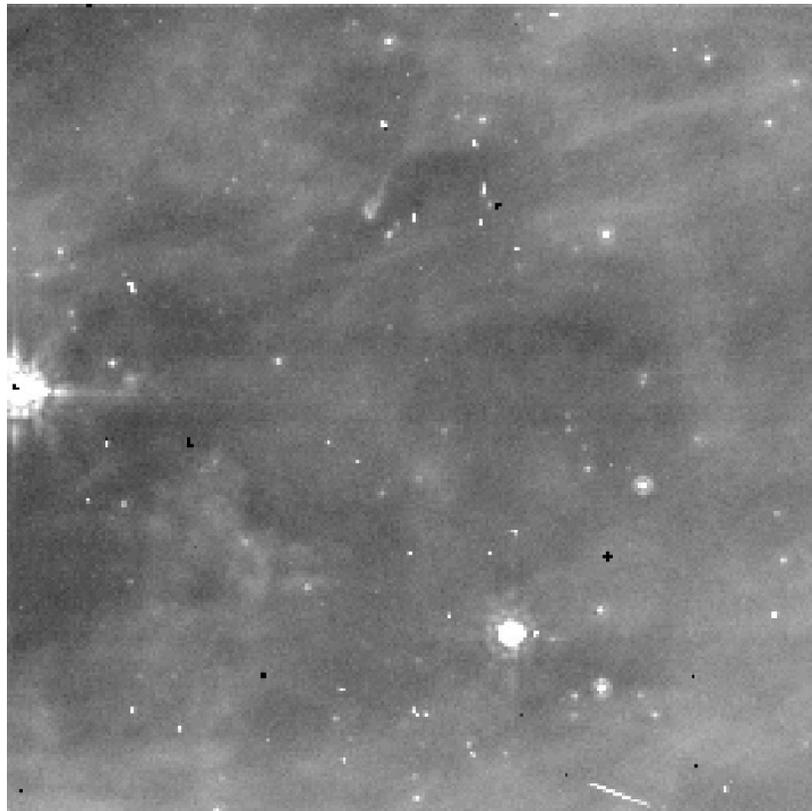


Figure 3. The image from Figure 2, after banding correction on both of the sources.

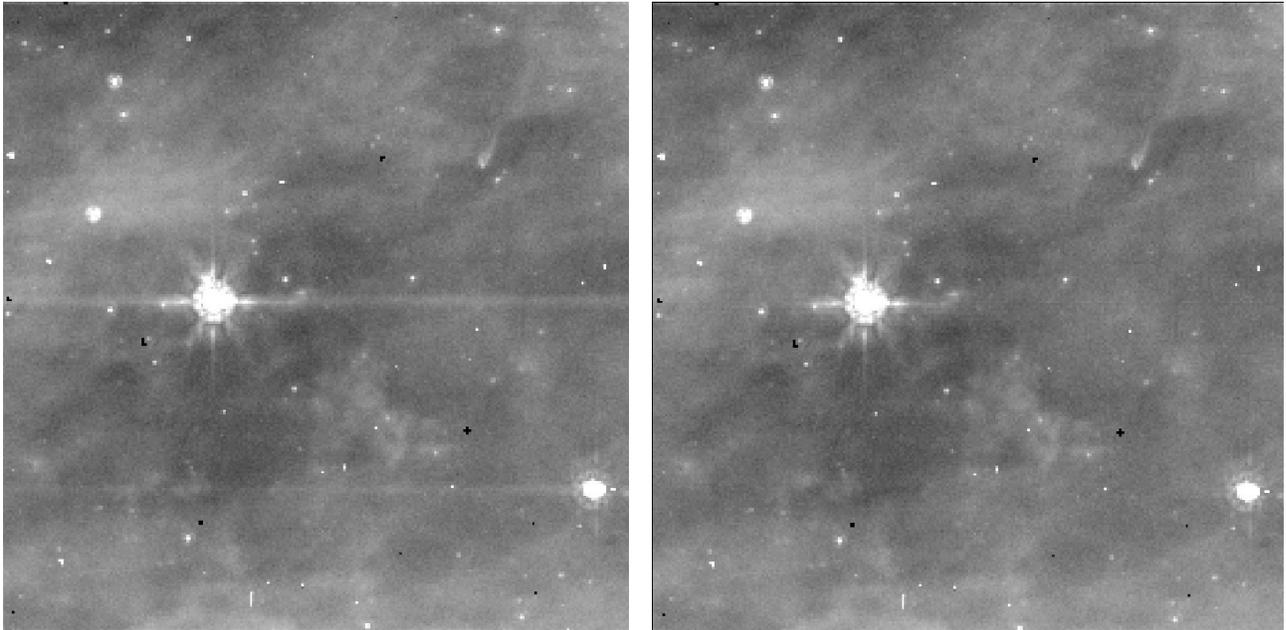


Figure 4. The next image in the list, with the same two sources shifted over slightly. The image on the left is before corrections, the image on the right is after corrections.

Most of the correction methods work by forcing the median of the columns or rows selected to match each other in the selected region, and that solution is applied to the whole row or column selected, or just the region in the box, depending on which command. To remove banding, for example, one selects a set of rows that contains a region of background around the banding that is large enough for the banding to be a smaller part of the area so that the artifact is “pulled down” to the median of the background. Other corrections, like the muxbleed or warm mission pulldown, apply a correction that varies along the row or column, and so are more sensitive to the location on the image and the range of pixels selected. See the sections below for a detailed description of each correction.

3.1. Banding correction (d or D command)

The banding correction works by calculating the median of the rows in the selected area. Then, for the “d” command, all rows with medians >0.5 sigma of the median for the full area are forced to have the median of the full area by adding or subtracting a constant to the full row. For the “D” command, all rows are forced to have the same median, so some correction is applied to all rows that are selected by the cursor commands. The “d” command is useful when there is a strong banding that is very obvious, and it removes the banding from the affected rows without changing other rows that are in the selection region. The “D” command is useful where the banding is weaker and it does not exceed the nearby noise level sufficiently for the “d” command to effectively remove the banding. One effective strategy is to first use the “d” command to remove strong banding, and then the “D” key to remove the small residual remaining.

After one is happy with the correction, one can use the “N” key to go to the next file in the list. The images from the list are displayed in order, with the first image in the lower left corner and

increasing to the right for the first 10 images, then moving up a row and starting from the left again for the next set of 10, etc. If one is not happy with the corrections, then the “u” key must be pressed before leaving the single image display, otherwise the changes remain permanent, and one would have to use the “o” command or otherwise retrieve the original files by hand.

The next image shows the same two sources, but now shifted over because of the mapping pattern being used in this AOR. The original file and corrected image is shown in Figure 4. After correcting the third image in this set, the “n” key was pressed to go back to the full display of all 100 frames, shown in Figure 5. The corrected images are now in this mosaic display, and one can then select the next images to work on.

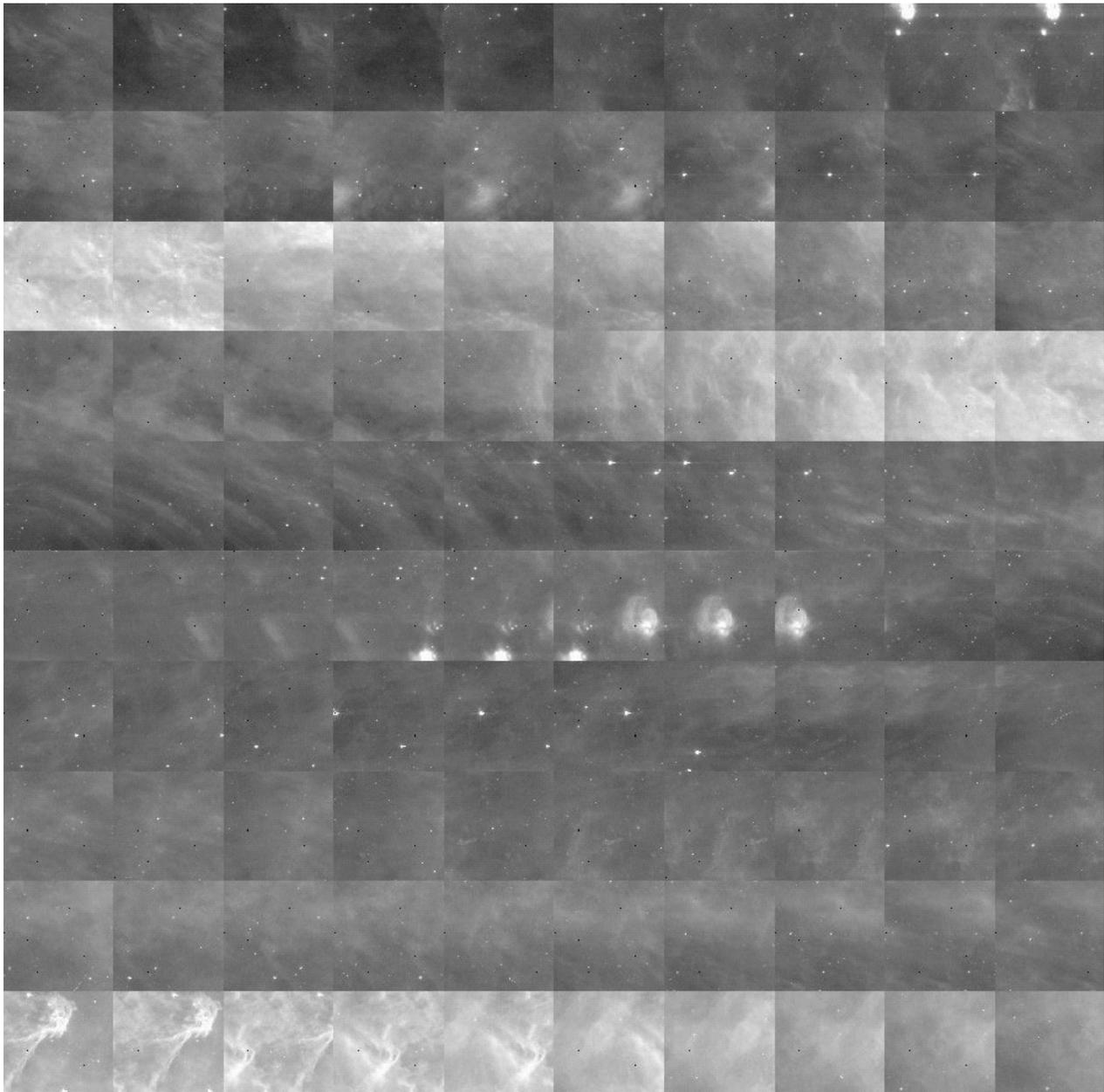


Figure 5. The full image display again, after corrections. The corrected versions of the files can now be seen, starting at the (4,4) position, relative to the lower left corner.

3.2. Column Banding correction (m or M command)

This command is similar to the row banding correction, but works on columns. This command is most useful for uncorrected column pulldown effects where the whole column has been affected by a bright source. The columns to act on are selected as in the banding correction, choosing a longer vertical region that includes the columns to be corrected plus sufficient background regions for the routine to correct the pulled-down columns (see Figure 6). Similar to the banding correction, the “M” key forces all columns to be adjusted, not just those that are greater than a certain sigma away from the median.

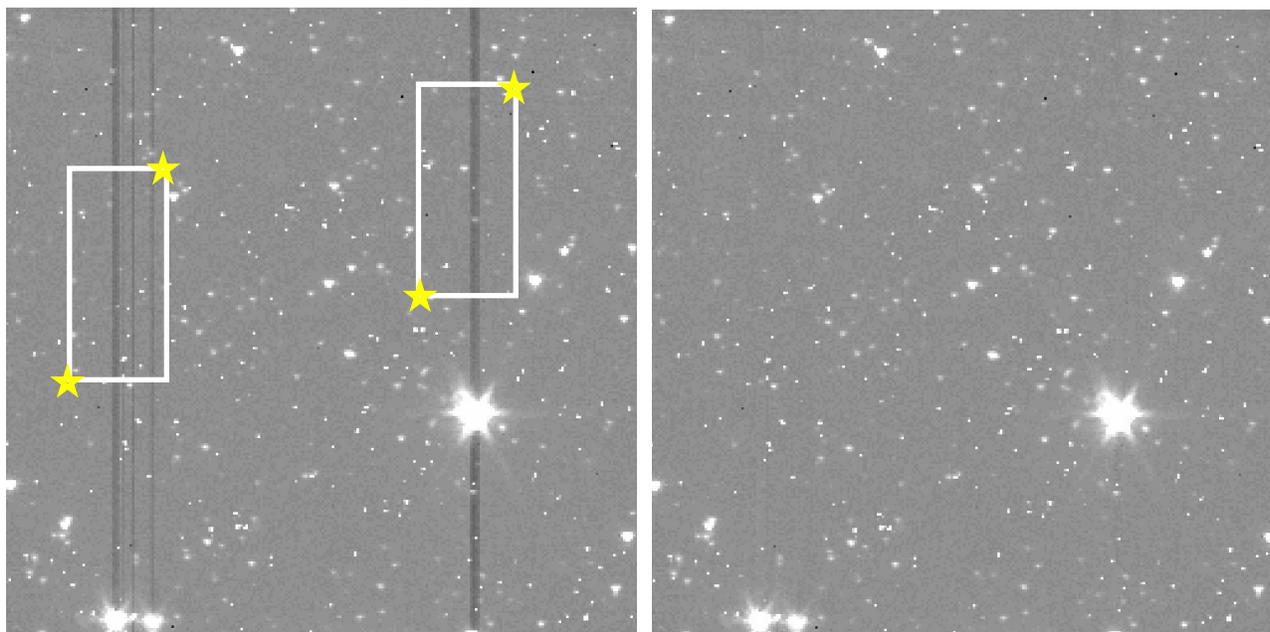


Figure 6. An example of vertical banding or “pulldown” corrections performed using the “m” command. Left: the regions are selected with two “m” commands for each region shown above (the yellow stars show the cursor locations where the “m” key was pressed). Right: the corrected image.

3.3. Warm mission column pulldown corrector (w command)

The column pulldown effect is different in the warm mission than in the cryo mission data. The main differences are that the pulldown level differs much more in the column below the bright star compared to the column above the star. Also, in the region below the star, the pulldown is not a simple constant value but varies along the column. This command is run similar to the “m” command described above, where a region is selected that contains the affected columns and a reference region around it. The program finds the affected columns and applies the appropriate corrections above and below the source. Because of the complexity of the artifact, the warm mission correction is often not as effective as the simpler cryo mission pulldown correction.

3.4. Column pulldown – section of column (v and V commands)

This command is similar to the m and M commands that correct column pulldown, except that it calculates the correction and applies it to only the selected part of the column, based on the region selected with the cursor keys when executing the command. This is useful in cases where for some reason the artifact is not uniform over the whole column. This sometimes happens when there are two bright stars in a single column. The correction can then be different in the regions above, between, and below the stars. Or this command could be where a single bright star has created a non-uniform column pulldown that varies along the column. Similar to the m and M commands described in Section 3.2, the v command only corrects columns that differ from the median value in the box by more than 0.5 sigma. The M command applies a correction to every column in the box. See Figure 7 for an example of this correction being used.

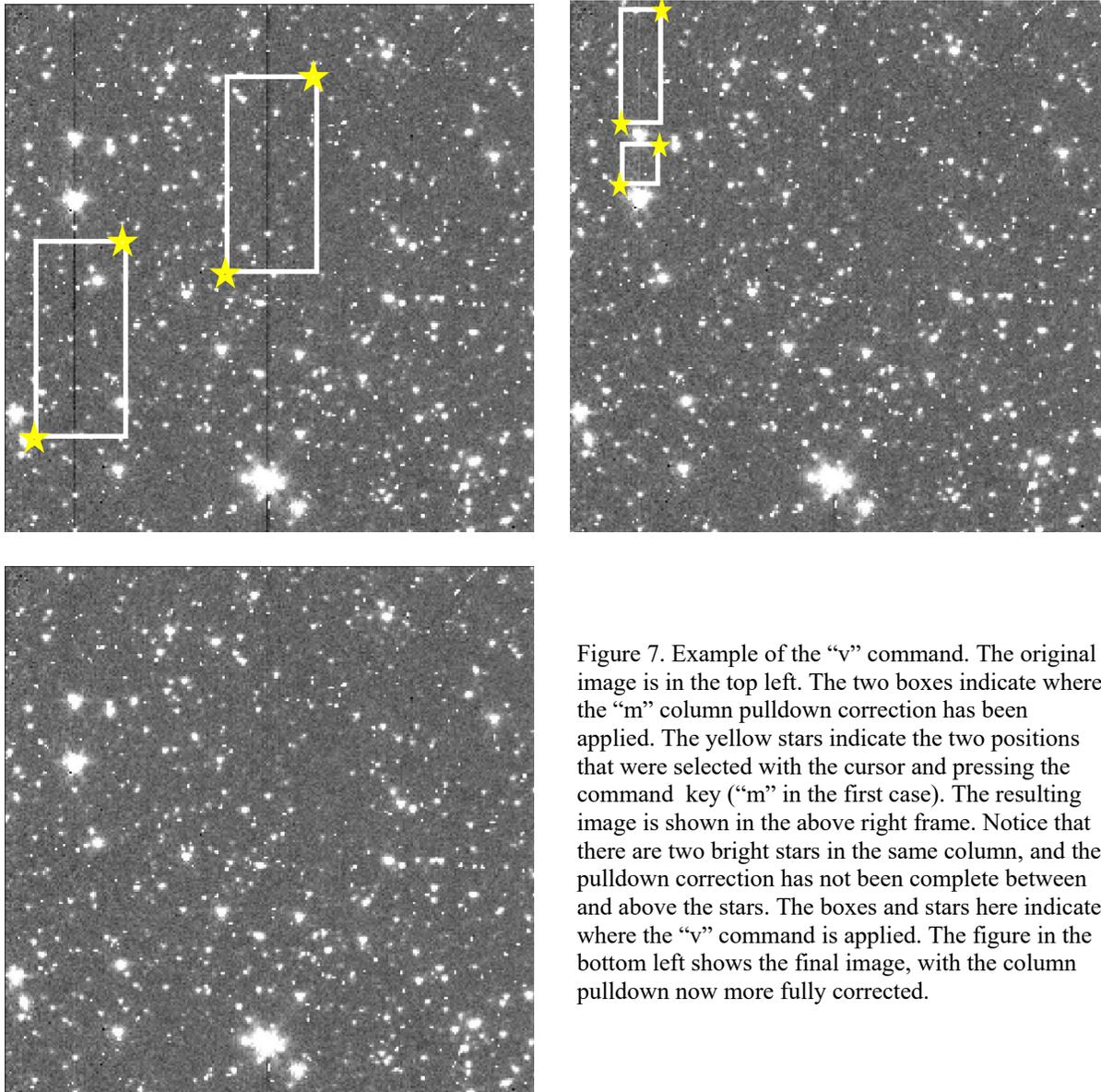


Figure 7. Example of the “v” command. The original image is in the top left. The two boxes indicate where the “m” column pulldown correction has been applied. The yellow stars indicate the two positions that were selected with the cursor and pressing the command key (“m” in the first case). The resulting image is shown in the above right frame. Notice that there are two bright stars in the same column, and the pulldown correction has not been complete between and above the stars. The boxes and stars here indicate where the “v” command is applied. The figure in the bottom left shows the final image, with the column pulldown now more fully corrected.

3.5. Row and column “pulldown” correction commands (c, l, r, k)

These commands work slightly differently than the others described above in that the region one selects is only the part of the image that is to be corrected, and the program will use a region outside of the selected area to use for the correction. For example, the “l” key will adjust the median of only the selected region, based on the columns immediately to the right of those selected. The “c” command is similar, except that it will apply the correction to the whole column(s), not just the selected region. Similarly, the “r” and “k” commands act on the partial and full rows selected. Even though these are called “pulldown” commands, the “m” and “M” commands are usually more effective in quickly removing pulldown effects, especially in cases such as shown in Figure 6 where there are multiple pulldown artifacts that can be corrected by a single command.

3.6. Masking regions (a command)

Sometimes the best strategy for eliminating artifacts such as stray light from bright sources outside of the field of view is to mask out a region so it is eliminated in the final mosaic. The SSC’s masks attempt to remove most of these artifacts, but there are often times when they are not completely removed. The artifacts are complicated in shape and hard to remove by fitting them, and can affect odd-shaped parts of the image (see Figure 8). One can select regions using the “a” command to write values to the *bimsk.fits file for the image being viewed. Then when the mosaics are made, the masked regions will not be used in the final image. The masking works best when the original observations used many frames to cover each mapping point. Obviously if there were not, the masking could lead to holes in the final image. The program sets the 2048 bit of the mask (0x800 hex) so the mosaicing program used to assemble the final image must also be configured so that pixels with this mask bit set are excluded from the mosaic. If your mask files do not end in “bimsk.fits”, you can change a parameter in imclean before running the script to match the names of your mask files (the parameter “masksuffix”).

3.7. Muxbleed correction (b or B command)

Muxbleed artifacts (an example is shown in Figure 9) are created by a bright star or cosmic ray in IRAC channel 1 or 2. This artifact is present in cryo mission data (taken prior to May 2009; data taken after that date sometimes has horizontal banding which can be corrected using the d or D commands). The muxbleed effect differs from the banding artifact in that there is a variation in the effect between the four IRAC readout channels, so that the banding is not smooth but has a pattern that repeats every 4th pixel as one goes from left to right across the array in the readout direction. The bleed decays in a nonlinear fashion from the bright pixel across the array to the right, and the effect “wraps around” to the next readout row, so therefore continues from left to right across the array in the next readout row. If the effect is caused by a bright star with many saturating pixels, each of these pixels causes a muxbleed effect which adds to the artifact, making it very difficult to model and subtract based on the magnitude of the offending star. The pixel phase of the star also determines how many pixels are involved and the shape of the artifact, so the pipeline correction is not always successful in completely removing the artifact from the BCDs.

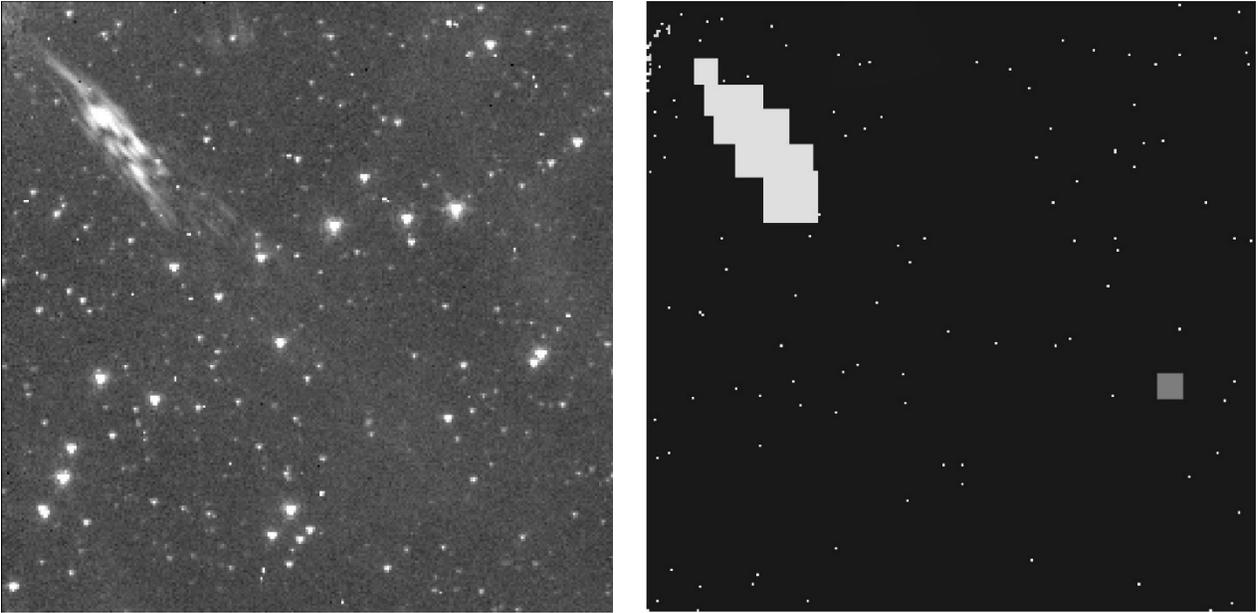


Figure 8. Masking regions. For the image on the left, several rectangular regions were defined to mask out the stray light feature seen in the upper left. The image on the right shows the resulting mask image, where one can see the newly masked regions as well as the other features that were previously in the mask.

The correction is usually done with the “B” command, which corrects all pixels within the selected region and to the right edge of the image in the selected rows. The “b” command could be used instead and will only correct every 4th pixel (one readout channel), starting with the first pixel on the left in the selected region. For this command, one selects the region in which to calculate the strength of the artifact. This should be a region affected by the muxbleed artifact, but not including bright sources that could affect the correction. The program assumes a pre-determined decay pattern of the muxbleed that was derived from a set of test images with stars of various brightnesses. The program fits this expected decay pattern to the data within the box to get the magnitude of the effect, using the region just below the box in the image to estimate the local background. This is done separately for each readout channel. Then the artifact is subtracted for pixels starting at the position of the box and continuing to the right all the way to the right edge of the image.

When correcting muxbleed for objects near the middle of the array as in Figure 9, one should start by correcting the artifact from the left side of the image to near the star. The correction will also be applied to the right edge of the image, so it will have some effect on the region to the right of the star, leaving some residual. Then one can correct the region to the right of the star to remove the residual.

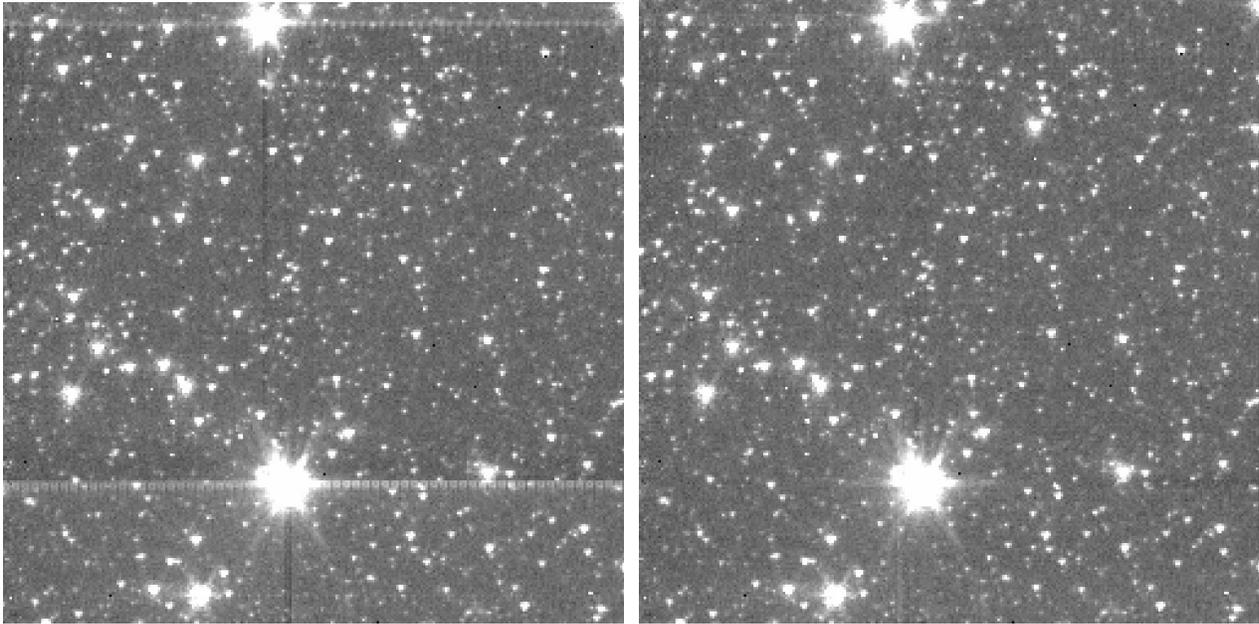


Figure 9. Muxbleed correction (B command). Left: image before correction. Right: after correction. Four separate commands were used to perform the correction, starting with the artifacts to the left of the bright stars, then correcting the residual artifacts on the right sides.

3.8. Striping correction (s and S commands)

Another artifact that is readout-channel dependent is the striping artifact, where vertical bars are seen across the image. The striping can be caused by bright sources in the image, especially if they fall near the edge of the array, or very bright stars anywhere in the image. An example is shown in Figure 10. Several bright stars have caused striping throughout the image. There is also extended emission that is brighter on the right side of the image. The “s” command allows you to sample a section of the image to determine the striping correction, which is then applied over the entire image. The correction simply calculates the median in each readout channel (every 4th column), and adds values to the lowest three of those readout channels to make the medians match the value for the highest median channel. Therefore, one should try to avoid image sections where there is a gradient in the background or a very bright object. In the example in Figure 10, the s command was used in the lower left quadrant in a region with no bright stars or extended emission. The second image in Figure 10 shows the result – the lower half of the image has been well-corrected, but there are residuals in the upper sections. If one examines the image closely, one sees that the striping is different in the regions between the two bright stars, compared to the smaller regions at the top of the image. In order to correct these regions, we can use the “S” command which calculates and corrections only the selected region. In most images like the example shown in Figure 10, one should select the region by moving the cursor outside of the image on the left and right sides, in order to ensure that all columns are selected. The first cursor entry should be outside the left edge and at the bottom of the region to be corrected, and the second entry should be outside the right edge of the image, at the top of the region to correct. In this example, the residual striping was corrected in three separate regions in the upper part of the image, since the striping seemed to change character in the different parts of the image delineated by the muxbleed from the bright objects. The resulting fully corrected image is shown in Figure 11.

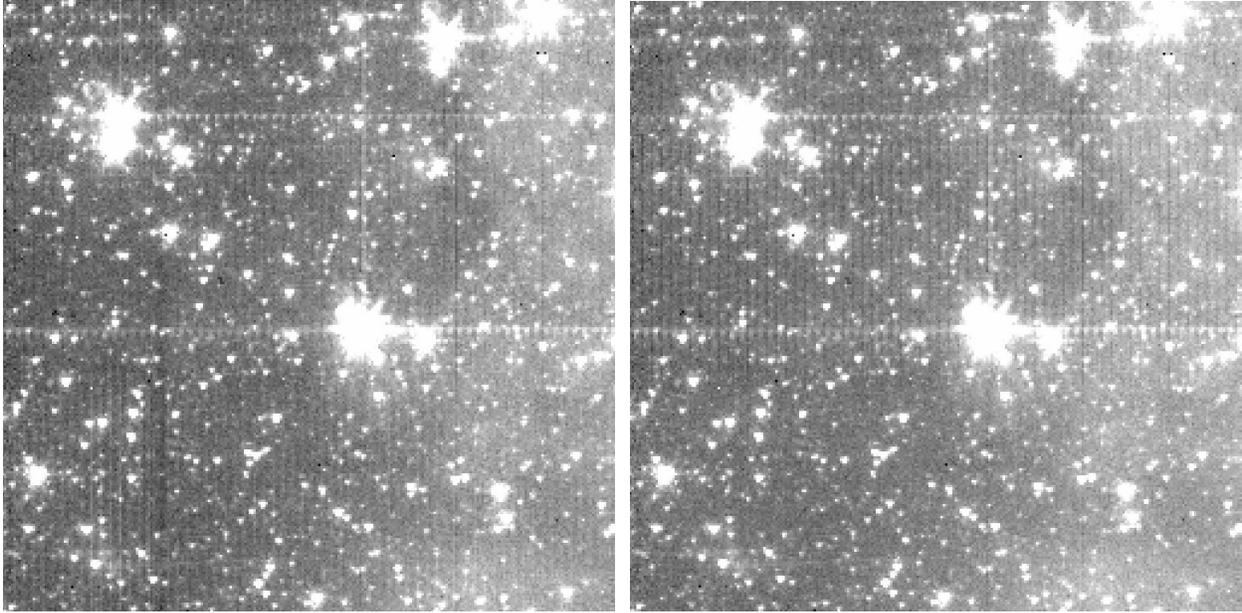


Figure 10. An example of striping artifacts in an IRAC channel 1 image. The left image has striping visible in all parts of the image. The image on the right has had the striping artifact removed from the lower half of the image using the “s” command in a region in the bottom left quadrant.



Figure 11. The same image as in Figure 10, except now the striping has been removed from the upper sections using the “S” command (the muxbleed was also correct as described in section 3.7).

3.9. Pixel interpolation (p or P command)

There are some circumstances where one might want to interpolate between pixels in an image. In general it is better to mask out pixels that have artifacts and allow the mosaicing process to fill in bad pixels or other effects. However, if there are cosmic rays or other effects that one wants to

remove from small regions, one can interpolate between pixels using the p or P commands. The only difference between these two is that after the “p” command, the display is not refreshed, whereas it is refreshed after the “P” command. Therefore the “p” command allows you to interpolate several pixels quickly without having to redraw the display each time.

The interpolation is performed between the nearest pixels outside of the region selected and applied to pixels within the region. Each row in the selected region is interpolated separately to pixels within that row. To interpolate only one pixel, the “p” key should be pressed twice without moving the cursor.

3.10. Overwrite file with original BCD file (o command)

This command copies the BCD file in the original AOR file directory and overwrites the file you are currently working on. Any changes you have made to the file will be lost. The main purpose of this command is to change from the cBCD version of the image to the BCD version. The SSC recommends that the cBCD version of the files be used in most cases, since these are the most highly processed and have the best artifact correction applied. However, there are some situations where the corrections applied have added significant artifacts or otherwise have degraded the image, and the BCD files are a better starting point for further corrections. This often happens in regions with bright extended emission where the artifact correction techniques try to estimate the image background but get confused by the structure in the image and overcorrects and otherwise further degrades the image, most often in Channels 3 and 4. An example is shown in Figure 12.

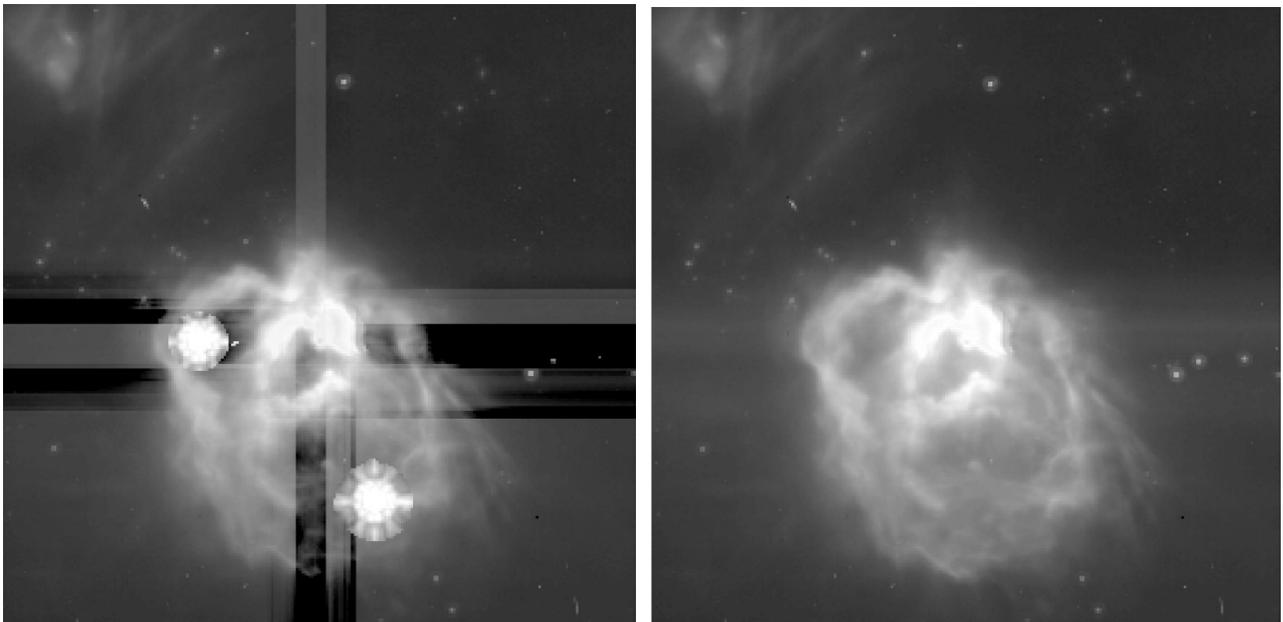


Figure 12. Example of an overcorrected cBCD file. **Left:** the cBCD image from the SSC pipeline in Channel 4 of a star forming region. The extended emission in the center of the image is saturated. The pipeline has detected this and attempted to remove the column pull-down and horizontal banding, but it has over- or undercorrected for the effect, leaving large horizontal and vertical bands. One can also see two spurious PSFs that it has for some reason added to locations in the extended emission. **Right:** the BCD version of the same frame. One can see that there is some slight horizontal banding from the bright emission, much less than in the cBCD version. Also, there is no sign of bright sources where the cBCD pipeline has inserted false PSFs.

Another artifact that primarily affects the Channel 3 and 4 cBCD files is that the pipeline attempts to locate saturated sources, and replace those pixels with a PSF scaled to what it determines the flux should be. However, sometimes the pipeline can be fooled by bright cosmic rays or features in the background, and insert random saturated PSFs in the image, as shown in Figure 12.



Figure 13. The BCD version of the file in Figure 12, corrected for banding with imclean. The central saturated region has not been modified; one would want to use the short frames from HDR mode to fill in these areas, or flag them in the final mosaic as being above the linear range of the detector.

In these cases, the best strategy is often to revert back to the BCD version and apply the necessary artifact corrections interactively with imclean. The “o” command will examine the current file name, determine what the original file name was, and attempt to copy it from the AOR directory given in the command line and overwrite the current file you are working on. The imclean script assumes that the files are either in the IRACproc format (if the parameter IRACproc is set to 1) or the original BCD or cBCD format (e.g., “SPITZER_I1_32700416_0028_0000_2_cbcd.fits”). The IRACproc format is similar but slightly abbreviated (e.g., “I1_32700416_0028_0_2_bcd.fits”). Once the file has been copied, the display is refreshed and the user can decide whether further corrections are necessary. The resulting corrected image for this example is shown in Figure 13.

4. spotclean – an imclean variant

A variant of `imclean` called `spotclean` has been added. This script is useful if you want to edit a subset of files around a certain position or object in the field. The format of the command is similar to `imclean`, here is an example:

```
cl> spotclean @filelist 334.701 57.903 0.1 -1 10 1 /data/PID1000/AORs/
```

where instead of the 100 image set number, you enter the RA, Dec of the position desired (in decimal degrees) and the half-size of the box (in degrees) to search. The program then searches your list of images for BCDs that are within the given distance from the position specified, and you can then operate the normal `imclean` routines on that smaller image set. Note that this simple search routine will have problems if your position is near 0 degrees RA and your search box extends into “negative” RA degrees, or if the Declination is near +90 or -90 degrees.

The `spotclean` script is useful for the cases where you have constructed your mosaic, but then you notice that there is some artifact remaining in the image that you want to correct. You can enter the position of that artifact in `spotclean`, and quickly edit the relevant BCDs to remove the feature, and then you can redo the mosaic to remove it from the final map. Another case that this can be useful is if you are really only interested in one object in the field, and don’t need to correct artifacts everywhere. You can enter your object coordinates in `spotclean`, and it will only display images that are relevant to your task.

`Spotclean` will only edit the first 100 images it finds within your box size, so if it displays 100 images to edit, there is a good chance that there are other images at this location that are not included in the display. If that is the case, you can just make your own custom list and use `imclean` to edit the files. For example, to create a file called “newlist” with all of the files within 0.1 degree of (334.701, 57.903), use the `iraf` command `hselect`:

```
cl> hselect @filelist $I "CRVAL1>334.601 && CRVAL1<334.801 && CRVAL2>57.803 && CRVAL2<58.003" > newlist
```

5. Change log (after 2019/07/23)

2019/12/10 – added the `v` and `V` functions

2020/02/26 – edits made to this document

2020/03/10 – installation instructions improved, typos fixed

2020/06/17 – edits made to this document for formatting and content