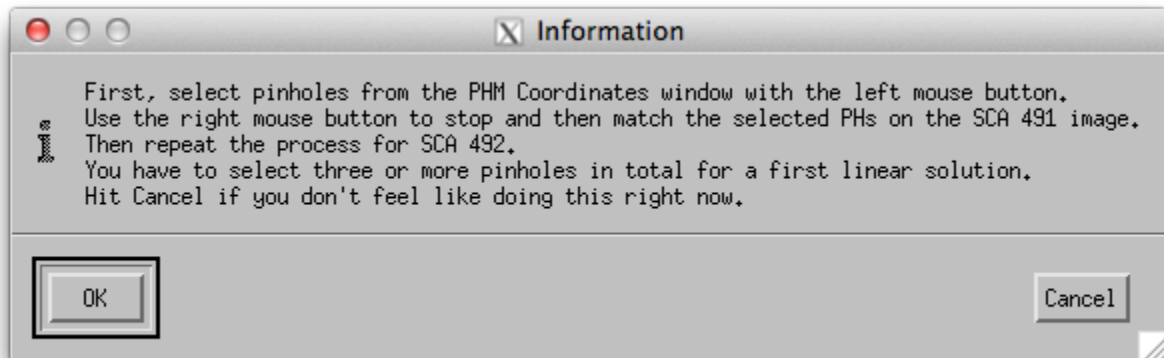
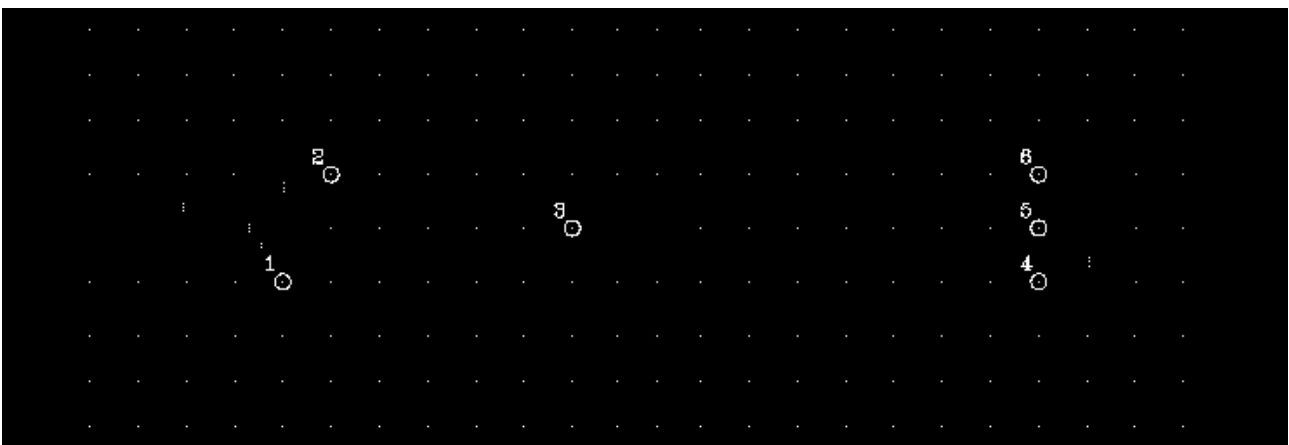


Here is a closer look at the instructions window:



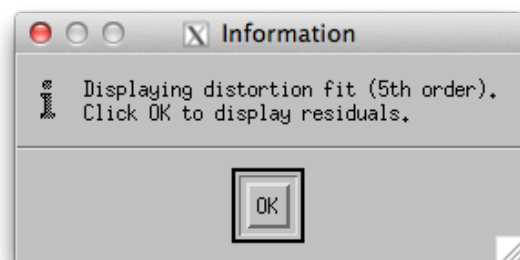
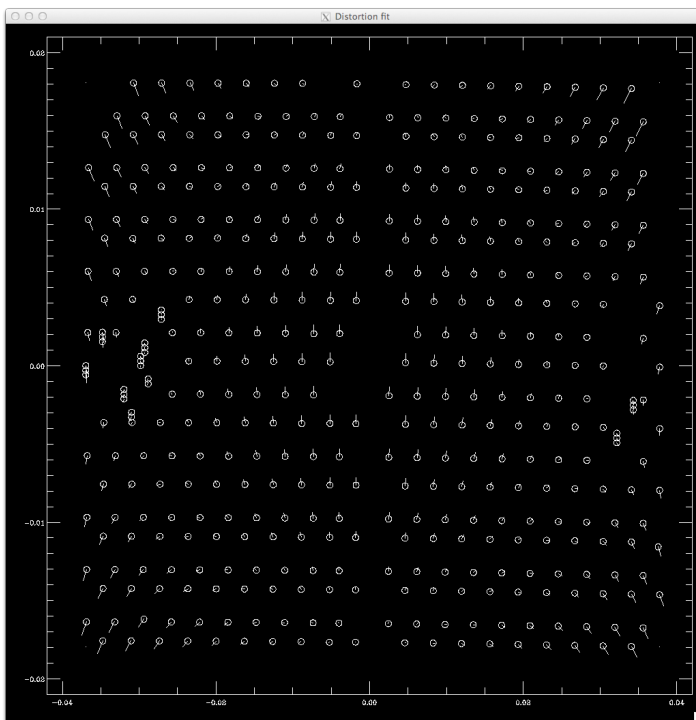
In the next steps, follow these instructions once you have dismissed the above window by clicking "OK". You can also stop the program at this point by clicking "Cancel" instead. Take a good look at the image on the left and the pinhole coordinates on the right. Are there patterns that can be easily identified to match pinholes? Select a few pinholes in the right window called "PHM Coordinates" with the left mouse button. Note the small circle annotated with a number that appears on the selected pinhole with each click. Hit the right mouse button once you are done. Please keep in mind that you need to identify at least three pinholes in total (both SCAs) in order to have enough data for a first linear transformation solution (offset, scale, and rotation/shear in x and y). Now, the left window named "SCA 491" is active, which can also be noted by the coordinates being displayed in the lower left corner. Left click on the pinhole images that match the ones selected in the other window. Again, a circle with a number will appear for every pinhole clicked. Once all are matched, the right window will become active again and the left window will show the count rate image taken with SCA 492. Now, repeat the above steps for the second detector. If you have already matched three or more pinholes in the steps above, you can also directly move on by clicking the right mouse button.

For this example I have selected six pinholes in total, three on each SCA. This is a zoom into the selected pinholes (PHM Coordinates window):

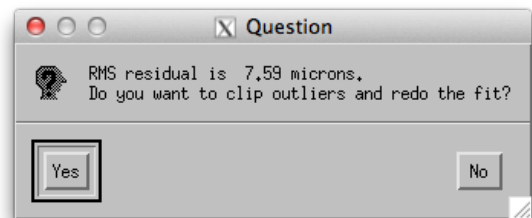
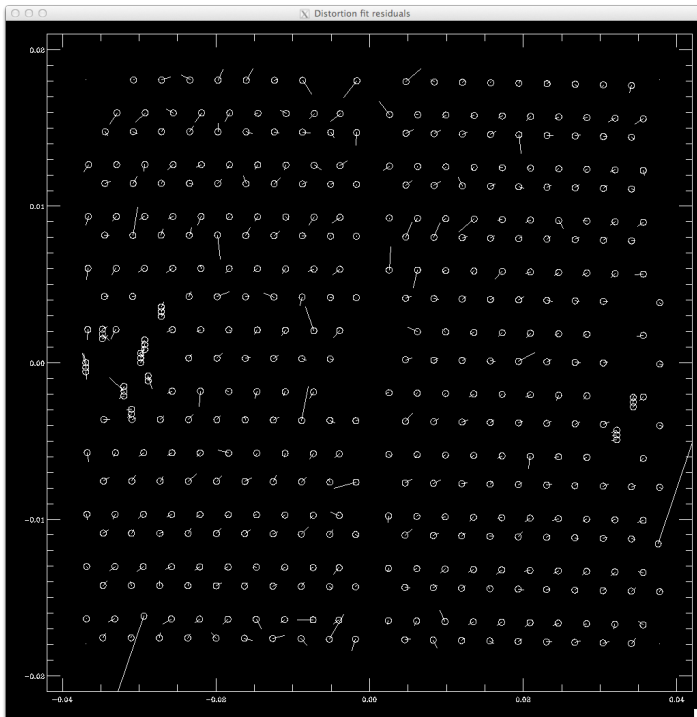


Once you are done matching pinholes, the program will try to match the remaining pinholes using the initial linear transformation solution computed from the pinholes you selected. In the left window you will note the SCA 491 exposure being displayed again. Large circles denote the predicted position of pinholes using the linear transform and PHM coordinates as input, smaller circles denote the actual position found. If a pinhole is crossed out it means that it was found and matched, but its signal-to-noise ratio is less than the required threshold (10 by default). The same will be done for SCA 492. If only very few pinholes get matched successfully and/or the estimated positions are tilted/stretched/fall in different detector areas, you probably made a mistake when matching the first pinholes manually in the step above. In this case just carry on following this guide and when done, try again. If there is more than one NID or set of exposures (in this example we use two sets), then the program will loop through the remaining files, displaying the same information as described above.

The program will then display the computed distortion for each pinhole position in the left window named "Distortion fit". The right window shows the deviation of each pinhole from its estimated position using the PHM coordinates and the linear transformation, i.e. the distortion. You will also get a message of what is displayed in the left window, including the order of the used distortion polynomial. Left window and message should look like this:



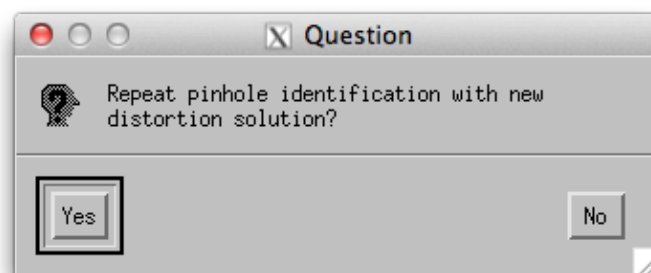
Click the "OK" button as the message suggests and the left image will display the residuals of the pinhole positions from the computed distortion fit. Also, a new message window will appear, showing you the root-mean-square (RMS) residual of all pinholes, and asking whether you would like to clip outliers. Plot and message window should look like this:



As there are some pinholes that are apparently more off than others, it might be a good idea to click “Yes” in this case. Do this and you will notice that a new message window telling you how many pinholes have been clipped. Furthermore, the clipped pinholes will be crossed out.

Once you acknowledge the message by clicking “OK”, a new linear transformation solution and distortion fit will be produced without the pinholes that were just clipped. Both plot windows will update, the left showing the distortion fit again, and you will get the message window with the order of the used distortion fit polynomial.

The above steps can be repeated as often as it is necessary. Eventually, you will be satisfied with the residuals or no pinholes will be clipped anymore. In either case, click “No” when you are asked about the clipping, and you will get the following message window:



This dialog asks whether you want to repeat the automatic pinhole identification/matching based on the determined transformation and distortion solution. Especially in case of significant distortion this can be very useful, because the program might not have found all

available pinholes because their true positions deviated too much from the predicted ones obtained with the initial linear transformation. In this example (ETU data) this is probably not necessary, but for simulated FM data using the IPS it was found to be the case.

Click “Yes” if you want to do so, the program will try to match the pinholes again (you do not have to select them by hand anymore) and the distortion fitting plus outlier clipping will take place as described above. Eventually you will end up with a refined distortion solution and this message window again. Click “No” to continue.

Now, you are basically done, the linear transformation and distortion solution has been computed, and it is time to decide what kind of output you would like to save.

First, you will encounter a dialog asking you whether you want to save the determined pinhole positions (on the SCAs, in pixel coordinates) to file. This file can be used as optional input to the `sb_derive_focus.pro` in order to speed up the selection of pinholes there (see Birkmann 2012 for details). As for the following queries, if you click “Yes” you will be prompted to select an output file name via a second dialog, otherwise you will continue to the next question.

The next question is whether you want to write the determined linear transformation solution and the distortion fit to file (forward and backward, in fits binary table format). This might be useful for other programs/tools in the future.

Finally, you will be asked if the program shall create a PDF file with the following plots: i) map of total distortion, ii) map of distortion in x-direction, iii) map of distortion in y-direction, iv) map of fitting residuals, v) point/vector representation of distortion as shown during execution, and vi) point/vector representation of the residuals.

Once you answered these questions and eventually saved (some) of the outputs, you are done and the program will end.

4 REFERENCES

- Birkmann, S. 2011, Installing the NIRSpec Calibration Software and Database on Mac OS X 10.7 (Lion), NIRSpec Technical Note NTN-2011-007, ESA/ESTEC
- Birkmann, S. 2012, Documentation for `sb_derive_focus.pro`, NIRSpec Technical Note NTN-2012-002, ESA/ESTEC
- Giardino, G. & Sirianni, M. 2011, NIRSpec Archive and Database, NIRSpec Technical Note NTN-2011-003, ESA/ESTEC

A REFERENCE GUIDE

Syntax | Arguments | Keywords | Examples

sb_distortion

The `sb_distortion` procedure allows the user to determine linear transformation and distortion from OTE to FPA plane by using images of pinholes taken with the CMO/PHM in place. This is achieved by fitting the position of pinhole images on the FPA and comparing them to the coordinates of the pinholes in the CMO/PHM.

Ideally, the used input images should be dark or background subtracted. This can be done relatively conveniently using the `NID` and `BKGD_NID` arguments/keywords.

If the procedure is called with the `NID` argument, it needs the `start_nirspec` environment and access to the NIRSpec data base and exposures.

This routine is written in the IDL language. Its source code can be found in the file `sb_distortion.pro` in the `Software/JWST_IDL/lib/nirspec/misc/` subdirectory of the NIRSpec Calibration Software and Database installation.

Syntax

```
sb_distortion [, nid] [, BKGD_NID=vector] [, IMG_PATH=string] [, ORDER=value] [, SNR=
value] [, ROT_ANGLE=value] [, IMG_OFFSET=value] [, IMG_SCALE=value] [, TRANS-
FORM={"DM_TRANSFORM" | "FM_TRANSFORM"}] [, CMO=array] [, /TA] [, /MOFFAT]
[, /CHEBYSHEV] [, /IFU] [, /SILENT] [, /VERBOSE]
```

Arguments

nid

An integer scalar or vector giving the NID(s) of the input exposures. The procedure uses the NIRSpec database to find and load the exposures. If not given the user will be prompted to select the input files via dialog.

Keywords

BKGD_NID

Set this keyword to an integer scalar or vector giving the NID(s) of the background/dark exposures to be subtracted from the input exposures. If shorter than the list of input exposures, the same background image will be used for several input images.

IMG_PATH

Set this keyword to a string giving the path to the input image files. If not given, the current directory is assumed. The user will only be prompted to select input files in the given path if the **nid** argument is not used, otherwise this keyword has no effect.

ORDER

Set this keyword to a scalar integer value between 2 and 5, specifying the order of the distortion polynomial. Default value is 5.

SNR

Set this keyword to a scalar value giving the signal-to-noise ratio threshold for fitted pinholes to be considered good. Only pinholes with higher SNR than specified are taken into account when determining transformation and distortion parameters. Default value is 10.

ROT_ANGLE

Set this keyword to a scalar value giving the rotation angle of the CMO/PHM coordinates with respect to the V-coordinate system in degrees. Default value is 221.5 degrees.

IMG_OFFSET

Set this keyword to a scalar value between 0 and 255 giving the grey value that the (clipped) mean of the input image will be displayed with. Default value is 50.

IMG_SCALE

Set this keyword to a scalar value greater not equal to zero giving the scaling of the input image from black to white in multiples of the (clipped) standard deviation of the input image displayed. Default value is 20. Use negative values for inverted scales (white to black).

TRANSFORM

Set this keyword to either "DM_TRANSFORM" or "FM_TRANSFORM". If given a pre-computed forward transformation will be read in from the PHM file and used to match the pinholes. You can still select pinholes manually at the beginning.

CMO

Set this keyword to a 3×3 array containing the transformation matrix for the CMO position to OTE coordinates. The default is to assume a sign change of both the x- and y-coordinates,

e.g. the transformation matrix:

$$CMO_{FW} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

TA

Set this keyword if you want the program to average the pinhole positions from exposures obtained with a small dither (less than one diagonal micro-shutter pitch). Intended use is for exposures obtained like during target acquisition (TA) with a diagonal dither of half a shutter pitch.

MOFFAT

Set this keyword in order to use a 2D Moffat to fit the pinhole images instead of the default 2D Gaussian.

CHEBYSHEV

Set this keyword in order to use Chebyshev polynomials for the distortion fit instead of the default standard polynomials.

IFU

Set this keyword in order to keep the pinholes located at the IFU aperture. By default, these pinholes are removed from the CMO/PHM list of pinholes prior automatic identification, because they are located at a different location on the FPA due to the virtual slits of the IFU.

SILENT

Set this keyword in order to suppress warning messages.

VERBOSE

Set this keyword in order to display some details when fitting the pinhole images.

Examples

This example will derive the transformation and estimate the distortion using the (DM) exposures with NIDs 2208 and 2210 which have been subtracted by the background exposures with the NIDs 2209 and 2211, respectively.

```
NIRSpec> sb_distortion, [2208,2210], BKGD_NID=[2209,2211]
```

This second example will use fifth order Chebyshev polynomials and require a signal-to-noise ratio of at least 25 for pinholes to be considered in the fit. The user will be prompted for the input files.

```
NIRSpec> sb_distortion, SNR=25, /CHEB
```