

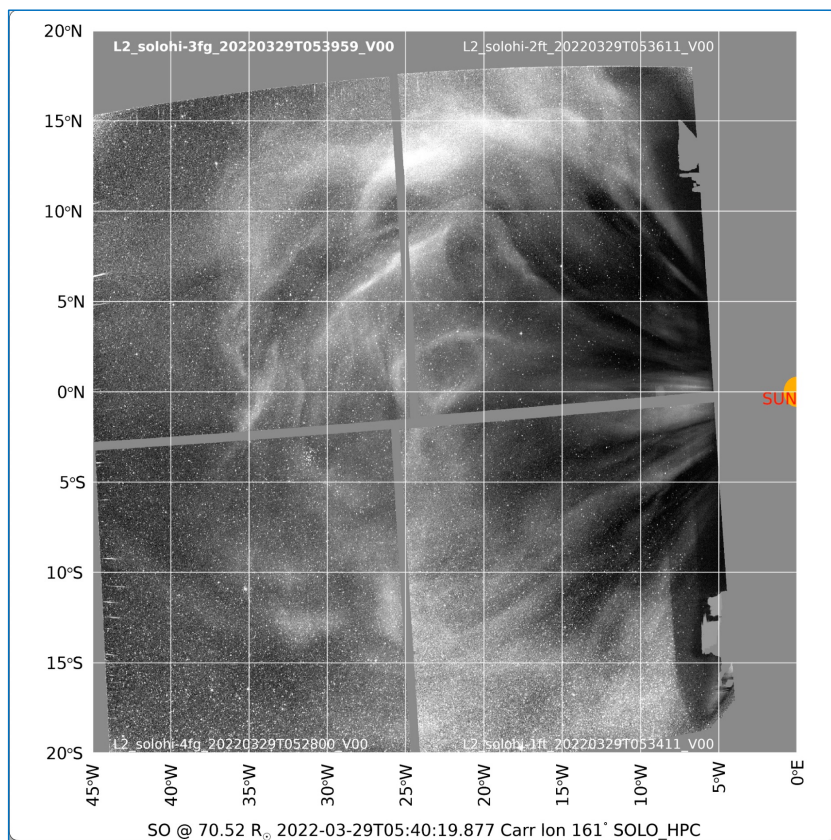
Background-Removed SoloHI Movie Process

P. Penteado (paulo.penteado@jpl.nasa.gov) & P. Liewer (paulett.c.liewer@jpl.nasa.gov)

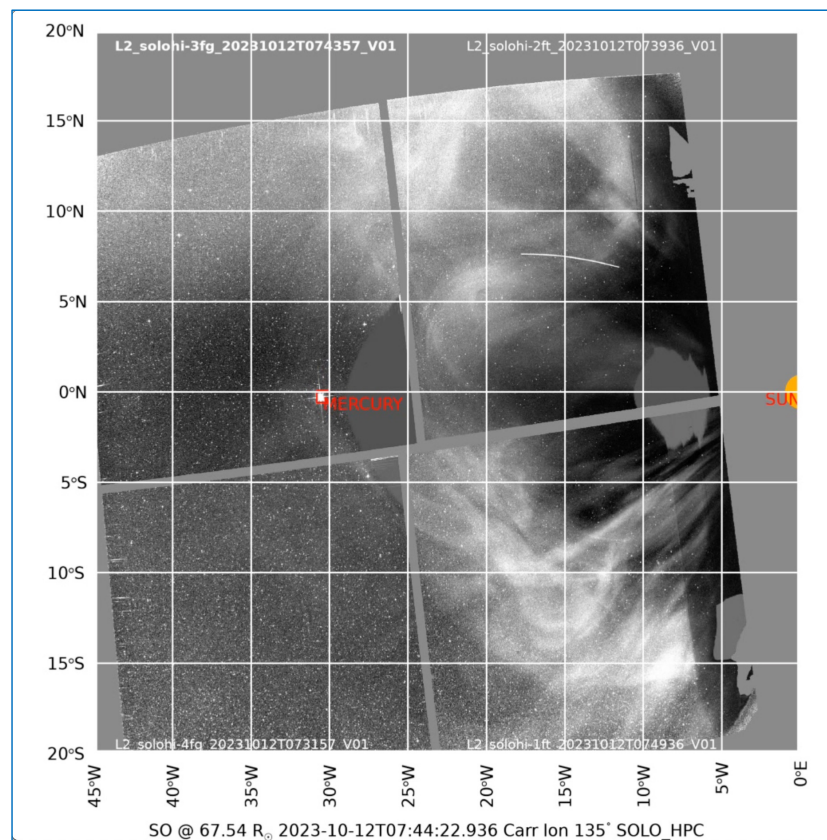
- Developed Technique for making background-removed full-FOV SoloHI images/movies
- Procedures to be available via SolarSoft-IDL along with scripts to make process easier
- Based on using image sequences from selected time intervals for which SC & solar arrays have not moved significantly. We call such a time intervals Sequences
 - Also creating timelines & tables of such Sequences [see next slide]
- For a given Sequence, creates backgrounds for each tile using percentiles & simple radial scaling with PSP distance from Sun. *Backgrounds only good for that Sequence*
- Also makes masks for Tiles 1 & 2 to cut out variable inner edge regions with stray light
- Divide image by its background to obtain background-removed image.
- Maps the 4 tiles to HelioProjective Cartesian (HPC) frame & creates pngs and movies
 - Unmapped individual tile FITS images created as a by-product

Sample Background-Removed HPC Movie Frames

2022 March 29 05:40 UTC



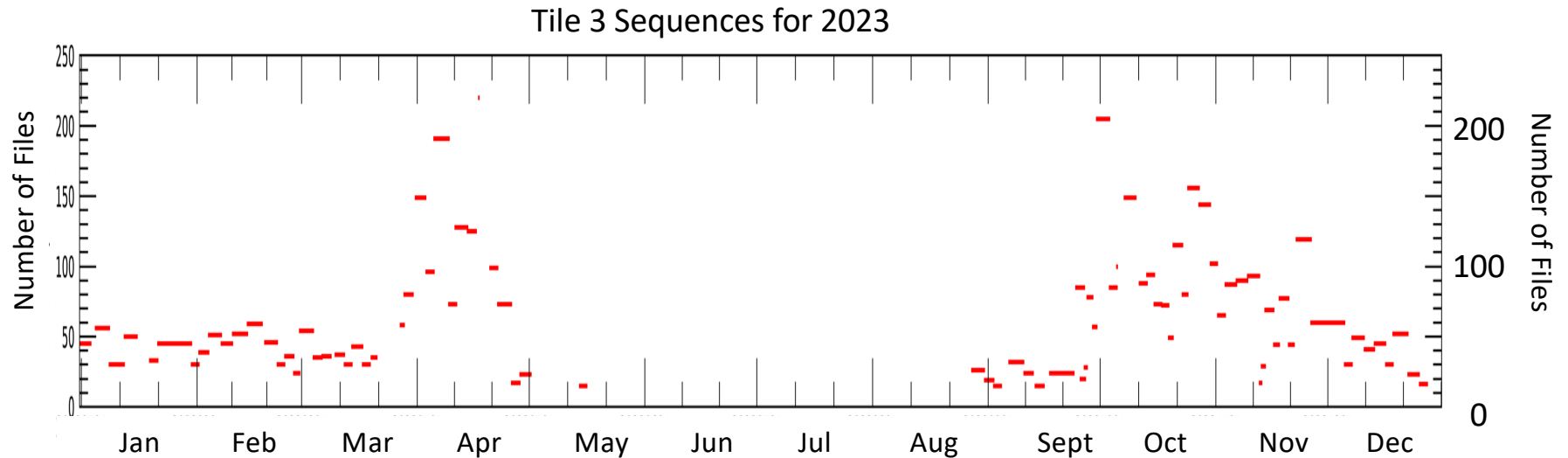
2023 October 12 07:44 UTC



Motivation: Timeline of Tile 3 Sequences for 2023

Sequence = Time intervals for which neither SC nor solar arrays have changed orientation

Red bar span length of the sequence; Y-axis the number of files in the sequence



Steps to Creating Background-removed images for a Sequence

Three Major Steps for Each Sequence

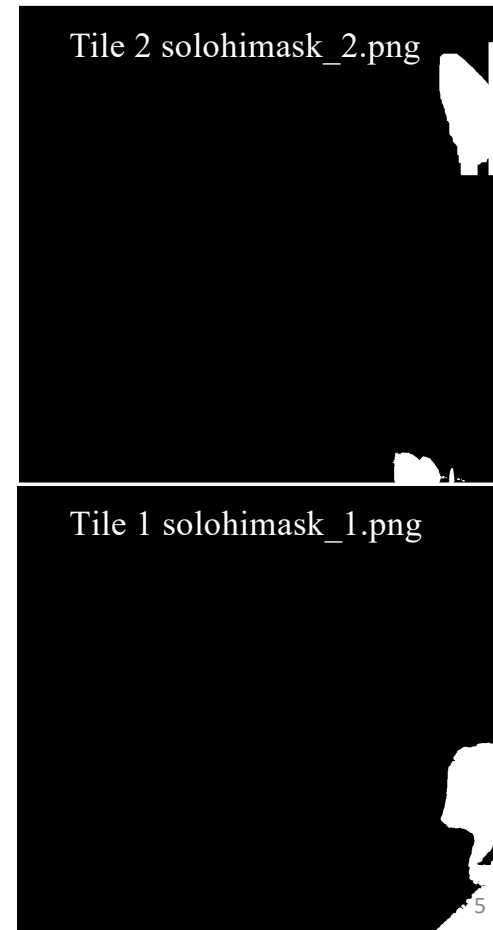
1. Make Bad Pixel Map for Tiles 1&2
2. Create backgrounds for the Sequence, one for each tile
3. Make background-removed images, both single tile and full FOV mapped images

1. Make Bad Pixel Maps* for Tiles 1&2

From Paulo's wiki page

- Bad pixel mask is computed from statistics on all images of that tile (1 or 2) in the sequence. These are put into a stack, which is used in several steps, to tag as bad pixels:
 - Plateaus (some have flat areas at the edges, a few pixels wide).
 - Regions where the pixel value goes over a threshold above the per-pixel median. Default is over the 97th percentile and those over 10 standard deviations above the mean. To avoid salt-and-pepper noise, this mask is convolved with a kernel to add any pixels that are neighboring those over the threshold.

* *Only Tiles 1&2 suffer from variable stray light from the solar arrays*

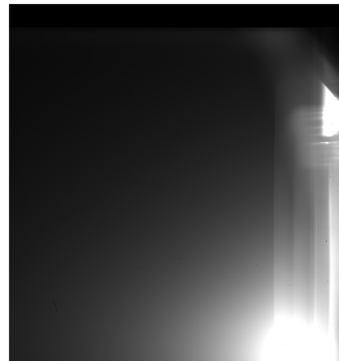


Step 2. Create backgrounds for the Sequence, one for each tile

1. Read all L2 FITS files for the time range of the sequence for the current tile (L2's are ~1k by 1k)
2. Apply normalization: dividing by R_p^2 , a constant value for the whole image, correcting for distance from Sun
3. Calculate a minimum value (1% level) for each pixel over all the images in the time sequence and create an image with this value for each pixel
4. Create the background image by filtering this image to remove occasional small problems such as dust streaks: The top 0.5% pixels are replaced with the average of pixels around them, calculated with a 5-pixel rectangle.
5. Save background images - cache or in an IDL Index Object

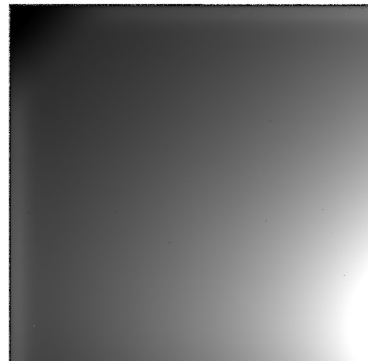


1

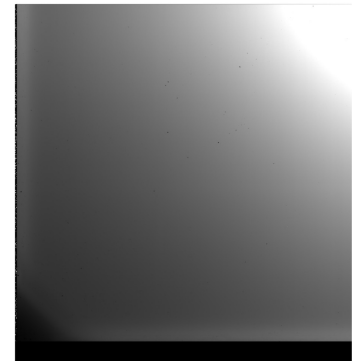


2

Sample Tile Backgrounds



3



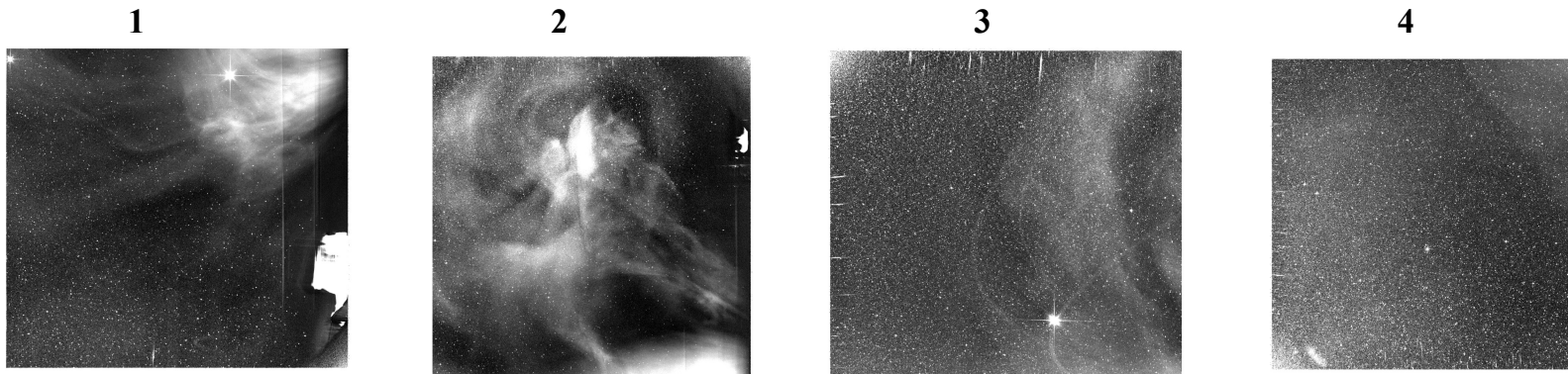
4

Step 3. Make background-removed images and maps (1/2)

Loop on each image from one of the tiles (tile 3, by default), doing:

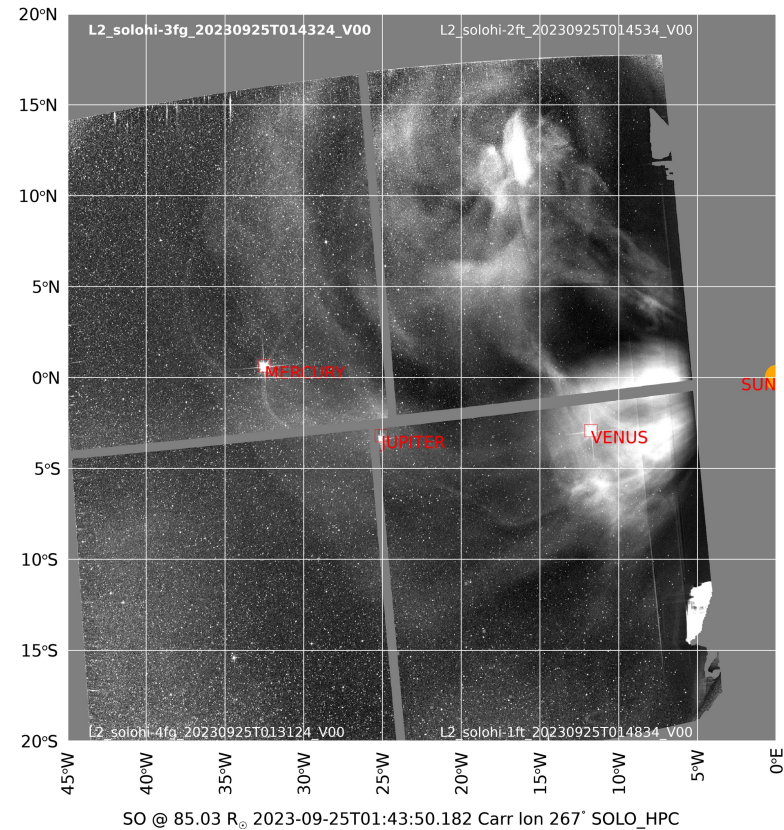
1. Find the 3 files for the other 3 tiles closest in time to go into the same final full FOV mapped images.
2. Apply R_p^2 normalization to the 4 images
3. Divide each image by its background
4. Apply bad pixel mask for tiles 1 and 2
5. Save the 4 resulting unprojected, background-removed, masked image as FITS
(pngs can also saved separately for each tile)

Background-Removed Tile Images

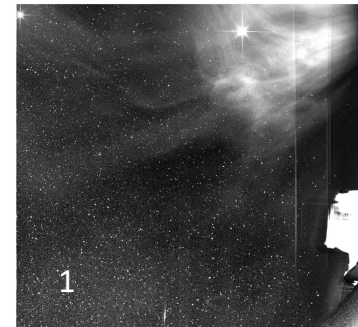
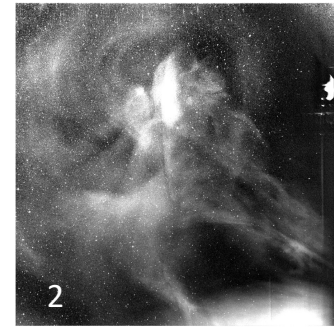
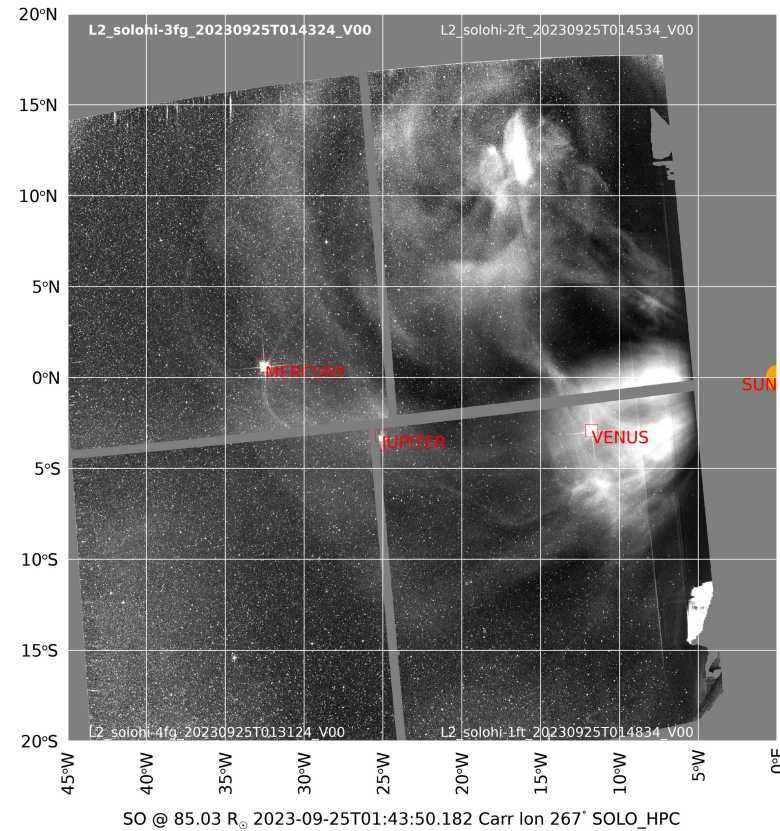
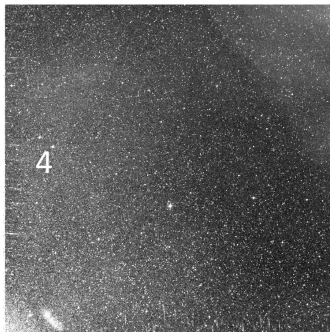
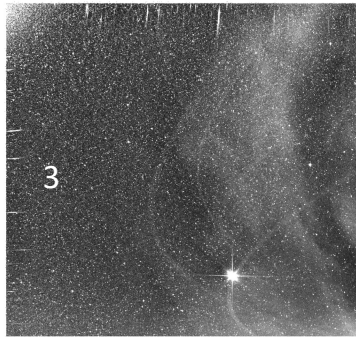


Step 3. Make background-removed images and maps (2/2)

6. Project each tile into the full FOV map – so only $\frac{1}{4}$ occupied by each tile.
7. Apply cut offs (defaults are 1% to 97.5% level) on projected tiles
8. Join all 4 mapped tiles into global map to produce one full FOV image - Joined image saved as FITS and PNG
9. Create movie from PNGs
10. *[Optional] Combine PNGs from adjacent Sequences to make longer movies*



Sample Tile pngs & final mapped full FOV image



2023 September 25T01:43

Movies to date (October 2024)

2022

solohi_20220325_20220403_T3_V31_ql.mp4

Event: 2022 March 25 CME

solohi_20220328_20220329_T3_V31_ql.mp4

Event: 2022 March 28 CME

2023

solohi_20230312_20230313_T3_V31_rel.mp4

Event: 2023 March 12 & 13 CMEs

solohi_20230421_20230425_v30_ql.mp4

Event: 2023 April 22 & 23 CMEs

solohi_20230924_20230926_T2_V30_ql.mp4

Event: 2023 September 24 CME

solohi_20230930_20231003_v30_ql.mp4

Event: 2023 October 1 & 2

solohi_20231010_20231014_T2_v30_ql.mp4

Event: 2023 October 10 & 12

2024

solohi_20240201_20240301_T3_v31_rel.mp4

All of February 2024/ every 4th T3 frame-
Combines Sequences

Sample scripts to make images and movies

1. Script to create an index of the files specified period of interest, divided into Sequences (`Solohi_build_index`) – Steps 1 &2
 - User inputs date range by day and path to SoloHI L2 files
 - Output is an Index Object `solohiindex_id` packed into an IDL save file
2. Script to make the movies using saved Index Object – Step 3
 - User inputs: name of save file with the Index Object `solohiindex_id`, time range for movie within time span of Index object, driving tile.....
 - Set a few keywords, e.g., to control whether or not to save pngs and/or FITS of individual tiles
 - Output is full FOV FITS and pngs and the individual frames as FITS/ pngs
 - Make pngs into a movie

Summary

- Have developed robust technique for making background removed SoloHI single-tile and full FOV images/fits/movies
- Technique based on finding Sequences within user specified time range (by day) in which SC & solar arrays have not moved significantly.
 - A background images is created for each of the 4 tiles for each sequence
- The information for the specified time period (by days), e.g., the sequences and all 4 backgrounds, are saved in IDL index objects
 - Saved because this is a very cpu intensive process
- Create background removed images, tile and full FOV images starting with an IDL index object for the period of interest
- The IDL Index Objects can be consider a separate data product

Sample scripts to make images and movies (1/2)

First, created an IDL index object which will have all the backgrounds for all the images in a specified range of days

IDL commands

```
;Set path to SoloHI L2 files - presumably has files by day in subfolders under L2/
L2path = '/export/solarraid1/ra/solohi/fm/rel/fits/L2/'
;Within this path, set date range for the folders that covers the time of interest
dirrange = ['20231008','20231016']
; Give a name for the file where the index object will be saved
fname1 = 'solohiindex_20231010.sav'
;Create metadata for the index object
SolohiindexA=solohi_index(fname1=fname1, root = L2path,dirrange=dirrange,verbose=2)
;Scan the directories and find the files in the specified range
solohiindexA.scan[,nt=4 ]           ;you probably want to use multiple threads, here
;Find sequence(s) and make & store the backgrounds and masks needed for all files
; You definitely want to use multiple threads here, but limit the number to be used
; because the default(all cores)can be too high
solohiindexA.findsequences [,nt=4 ]
;Check basic information on the index object, e,g., number of files, first&last file
; & name of save file
help, solohiindexA
```

Sample scripts to make images and movies (2/2)

Now, make the background-removed images and mapped full FOV images using a saved index object

IDL commands

```
;Restore an index object (contains a list of files, sequences and all backgrounds &
masks for the specified days)
  solohiindexB=solohi_index(restore= 'solohiindex_20231010.sav')
;Check what days are in the object
  help, solohiindexB
;Get the list of files in the object
  su=solohiindexB.summary
;Chose a tile to determine the cadence of the full FOV movie
  tile = 3 ; the default
;Specifiy time range of movie (within days of object) &find of all tile = 3 files in
that range using FITS keywords
  w=where((su.wcs.time.observ_avg ge '2023-10-10T10:00:00:00') and $
    (su.wcs.time.observ_avg le '2023-10-14T00:00:00:00')and(su.detector eq tile))
;Make a list of the tile=3 files
  fs=su[w].fname
;Run the pipeline: find files to go with tile=3 files, divide by backgrounds etc.*
  solohi_join_caller_p,fs=fs,root=root,nt=nt,savel2=savel2,$
  index=solohiindexB,sys='SOLO_HPC',noclobber=noclobber,l2only=l2only,/fitsfile
;Turn pngs into a movie
  pp_pngtvideo,dir=root

*savel2, sys, noclobber,l2only are various options
```

Motivation: Timeline of Sequences for 2023 in Table Form

Intervals for which neither SC nor solar arrays have changed orientation & number files

🔗 Year 2023, ql, detector 2

DETECTOR	SEQUENCE	NF	FNAME_MIN	FNAME_MAX
2	0	96	solo_L2_solohi-21s_20230927T022836_V00	solo_L2_solohi-21s_20230927T053836_V00
2	1	119	solo_L2_solohi-2ft_20231218T002356_V00	solo_L2_solohi-2ft_20231221T224756_V00
2	2	59	solo_L2_solohi-2ft_20230430T002420_V00	solo_L2_solohi-2ft_20230503T193624_V00
2	3	297	solo_L2_solohi-2ft_20230401T002411_V00	solo_L2_solohi-2ft_20230403T180011_V00
2	4	233	solo_L2_solohi-2ft_20231106T002324_V00	solo_L2_solohi-2ft_20231109T233528_V00
2	5	204	solo_L2_solohi-2ft_20231031T011550_V00	solo_L2_solohi-2ft_20231102T160354_V00
2	6	184	solo_L2_solohi-2ft_20231017T122737_V00	solo_L2_solohi-2ft_20231019T200338_V00

NF = Number of files in the sequence

FNAME_MIN = name of first file

FNAME_MAX = name of last file