

Astrometry

Links to Astrometry and WCS References:

[More information on WCS than you ever wanted to know.](#)

It's been surprisingly difficult to get astrometry.net to latch onto the images we produce. This is presumably due to the wide field and distortion.

For one object catalog I did get a solution, which indicates the center of the image is pointed at dec of -28.7. This was for image

ut011314.0360.long.fits

which gave

RA, Dec = (109.299, -28.7017), pixel scale 169.598 arcsec/pix.

here is the corresponding wcs.fits file:

[wcs.fits](#)

FITS images are N up E left!

and here it is listed out:

```
cp65239:Downloads cstubbs$ imhead wcs.fits
SIMPLE =                T / Standard FITS file
BITPIX =                8 / ASCII or bytes array
NAXIS =                 0 / Minimal header
EXTEND =                T / There may be FITS ext
WCSEXES =               2 / no comment
CTYPE1 = 'RA---TAN-SIP' / TAN (gnomic) projection + SIP distortions
CTYPE2 = 'DEC--TAN-SIP' / TAN (gnomic) projection + SIP distortions
EQUINOX =              2000.0 / Equatorial coordinates definition (yr)
LONPOLE =              180.0 / no comment
LATPOLE =               0.0 / no comment
CRVAL1 =   109.320631076 / RA  of reference point
CRVAL2 =  -28.6668823198 / DEC of reference point
CRPIX1 =           1192 / X reference pixel
CRPIX2 =           960.5 / Y reference pixel
CUNIT1 = 'deg'        ' / X pixel scale units
CUNIT2 = 'deg'        ' / Y pixel scale units
CD1_1 =   -0.047513171544 / Transformation matrix
CD1_2 =   -0.00146819731348 / no comment
CD2_1 =   -0.000864857162952 / no comment
CD2_2 =    0.0466847846798 / no comment
IMAGEW =           2382 / Image width,  in pixels.
IMAGEH =           1919 / Image height, in pixels.
A_ORDER =                2 / Polynomial order, axis 1
A_0_2 =   4.6078119764E-06 / no comment
A_1_1 =   1.23847648274E-05 / no comment
```

A_2_0 = -5.29280627903E-06 / no comment
B_ORDER = 2 / Polynomial order, axis 2
B_0_2 = 4.65369941262E-06 / no comment
B_1_1 = 2.97633768011E-05 / no comment
B_2_0 = -5.10163633601E-05 / no comment
AP_ORDER= 2 / Inv polynomial order, axis 1
AP_0_1 = -9.0981000082E-05 / no comment
AP_0_2 = -4.58575725434E-06 / no comment
AP_1_0 = -0.000380322057888 / no comment
AP_1_1 = -1.23836775631E-05 / no comment
AP_2_0 = 5.29784239136E-06 / no comment
BP_ORDER= 2 / Inv polynomial order, axis 2
BP_0_1 = -0.000401626462877 / no comment
BP_0_2 = -4.59507257312E-06 / no comment
BP_1_0 = -0.000786806957891 / no comment
BP_1_1 = -2.97521982273E-05 / no comment
BP_2_0 = 5.09931822201E-05 / no comment
HISTORY Created by the Astrometry.net suite.
HISTORY For more details, see <http://astrometry.net> .
HISTORY Subversion URL
HISTORY <http://astrometry.net/svn/tags/nova/2013-05-24-2/util/>
HISTORY Subversion revision 22686
HISTORY Subversion date 2013-05-06 22:33:22 +0000 (Mon, 06 May
HISTORY 2013)
HISTORY This WCS header was created by the program "blind".
DATE = '2014-01-14T04:03:54' / Date this file was created.
COMMENT -- blind solver parameters: --
COMMENT Index(0): /data1/INDEXES/200-current/index-219.fits
COMMENT Index(1): /data1/INDEXES/200-current/index-218.fits
COMMENT Index(2): /data1/INDEXES/200-current/index-217.fits
COMMENT Index(3): /data1/INDEXES/200-current/index-216.fits
COMMENT Index(4): /data1/INDEXES/200-current/index-215.fits
COMMENT Index(5): /data1/INDEXES/200-current/index-214.fits
COMMENT Index(6): /data1/INDEXES/200-current/index-213.fits
COMMENT Index(7): /data1/INDEXES/200-current/index-212.fits
COMMENT Index(8): /data1/INDEXES/200-current/index-211.fits
COMMENT Index(9): /data1/INDEXES/200-current/index-210.fits
COMMENT Index(10): /data1/INDEXES/200-current/index-209.fits
COMMENT Index(11): /data1/INDEXES/4100/index-4119.fits
COMMENT Index(12): /data1/INDEXES/4100/index-4118.fits

COMMENT Index(13): /data1/INDEXES/4100/index-4117.fits
COMMENT Index(14): /data1/INDEXES/4100/index-4116.fits
COMMENT Index(15): /data1/INDEXES/4100/index-4115.fits
COMMENT Index(16): /data1/INDEXES/4100/index-4114.fits
COMMENT Index(17): /data1/INDEXES/4100/index-4113.fits
COMMENT Index(18): /data1/INDEXES/4100/index-4112.fits
COMMENT Index(19): /data1/INDEXES/4100/index-4111.fits
COMMENT Index(20): /data1/INDEXES/4100/index-4110.fits
COMMENT Index(21): /data1/INDEXES/4100/index-4109.fits
COMMENT Field name: job.axy
COMMENT Field scale lower: 15.1134 arcsec/pixel
COMMENT Field scale upper: 272.04 arcsec/pixel
COMMENT X col name: X
COMMENT Y col name: Y
COMMENT Start obj: 0
COMMENT End obj: 0
COMMENT Solved_in: (null)
COMMENT Solved_out: (null)
COMMENT Solvedserver: (null)
COMMENT Parity: 2
COMMENT Codetol: 0.01
COMMENT Verify pixels: 1 pix
COMMENT Maxquads: 0
COMMENT Maxmatches: 0
COMMENT Cpu limit: 600.000000 s
COMMENT Time limit: 0 s
COMMENT Total time limit: 0 s
COMMENT Total CPU limit: 0.000000 s
COMMENT Tweak: yes
COMMENT Tweak AB order: 2
COMMENT Tweak ABP order: 2
COMMENT --
COMMENT -- properties of the matching quad: --
COMMENT index id: 4116
COMMENT index healpix: -1
COMMENT index hpnside: 0
COMMENT log odds: 1.6833
COMMENT odds: 5.38327
COMMENT quadno: 2848
COMMENT stars: 3597,3254,3606,0

```

COMMENT field: 67,52,32,0
COMMENT code error: 0.00321729
COMMENT nmatch: 4
COMMENT nconflict: 1
COMMENT nfield: 200
COMMENT nindex: 820
COMMENT scale: 169.598 arcsec/pix
COMMENT parity: 0
COMMENT quads tried: 3410976
COMMENT quads matched: 3383383
COMMENT quads verified: 156020
COMMENT objs tried: 68
COMMENT cpu time: 67.2642
COMMENT --
END

```

but that might be enough to get us started....:

```
cp65239:Downloads cstubbs$ xy2sky -d wcs.fits 1000 1000
```

```
119.38013 -26.40007 J2000 1000.000 1000.000
```

SO let's take the SIMBAD catalog and transform into coordinates for this image.

```
526 sky2xy 360.wcs.fits @Simbad.radec.dat | grep -v "off" > Simbad.xy.dat
```

```
527 more Simbad.xy.dat
```

```
528 awk '{print $5, $6}' Simbad.xy.dat
```

```
529 awk '{print $5, $6}' Simbad.xy.dat > xy.dat
```

```
530 ds9mark.pl ut011314.0806.long.fits xy.dat
```

Yuk.

Seems almost OK at the center but rapidly degrades as you move out.

image was taken at UT Jan 14 03:39 which, according to iObserve, corresponds to LMST=06:30 or in decimal degrees it's 97.5 degrees. That's a little off from the nominal center of the image being 109.35, but maybe it isn't pointed straight up?

Tried astrometry.net with initial center guess of 97.5 and -30, with third order polynomial fit. It nailed it instantly!

RA of 96.00 and dec of -28.354

Upload Settings	
Parity:	try both simultaneously
Scale Units:	arcseconds per pixel
Scale Type:	bounds
Scale Lower Bound:	150.0
Scale Upper Bound:	190.0
Star Positional Error:	5.0 pixels
Limits:	RA, Dec - (97.5, -30.0) radius - 10.0

Downsample Factor:	2
Tweak Order:	3
Source detection:	SExtractor
CRPIX:	center

It does seem the field center is correct, but there are bad mis-matches between the Simbad catalog and the objects in the field, gets worse away from the center.

Try with out own photometric catalog, and matching to a subsequent image.

get image 460:

```
cp65239:calib cstubbs$ scp christopherstubbs@139.229.19.220:/Volumes/2TB/ut011314/ut011314.0460.long.cr2.gz .
```

```
cp65239:calib cstubbs$ cr2fits -bw ut011314.0460.long.cr2
```

```
DATE-OBS= '2014-01-14T04:41:24' // Exposure date
```

so this is one hour and two minutes later than image 360, which should add 15.5 degrees in RA, or 111.5 RA and -28.354 in dec. Try astrometry.net for that one.

Yup, nailed that one too. This is not so bad.

Center (RA, Dec):	(111.372, -28.307)
Center (RA, hms):	07 ^h 25 ^m 29.341 ^s
Center (Dec, dms):	-28° 18' 25.392"
Size:	135 x 89.9 deg
Radius:	81.055 deg
Pixel scale:	168 arcsec/pixel

So let's take photometry catalog from image 360 and transform into pixel coords of image 460.

```
579 sex ut011314.0360.long.fits
```

```
580 more test.cat
```

```
581 grep -v '#' test.cat | awk '{print $9, $10}'
```

```
582 grep -v '#' test.cat | awk '{print $9, $10}' > 360.xy.cat
```

```
583 xy2sky image.360.wcs.fits @360.xy.cat
```

```
584 xy2sky -d image.360.wcs.fits @360.xy.cat
```

```
587 xy2sky -d image.360.wcs.fits @360.xy.cat | awk '{print $1, $2}' > 360.radec.cat
```

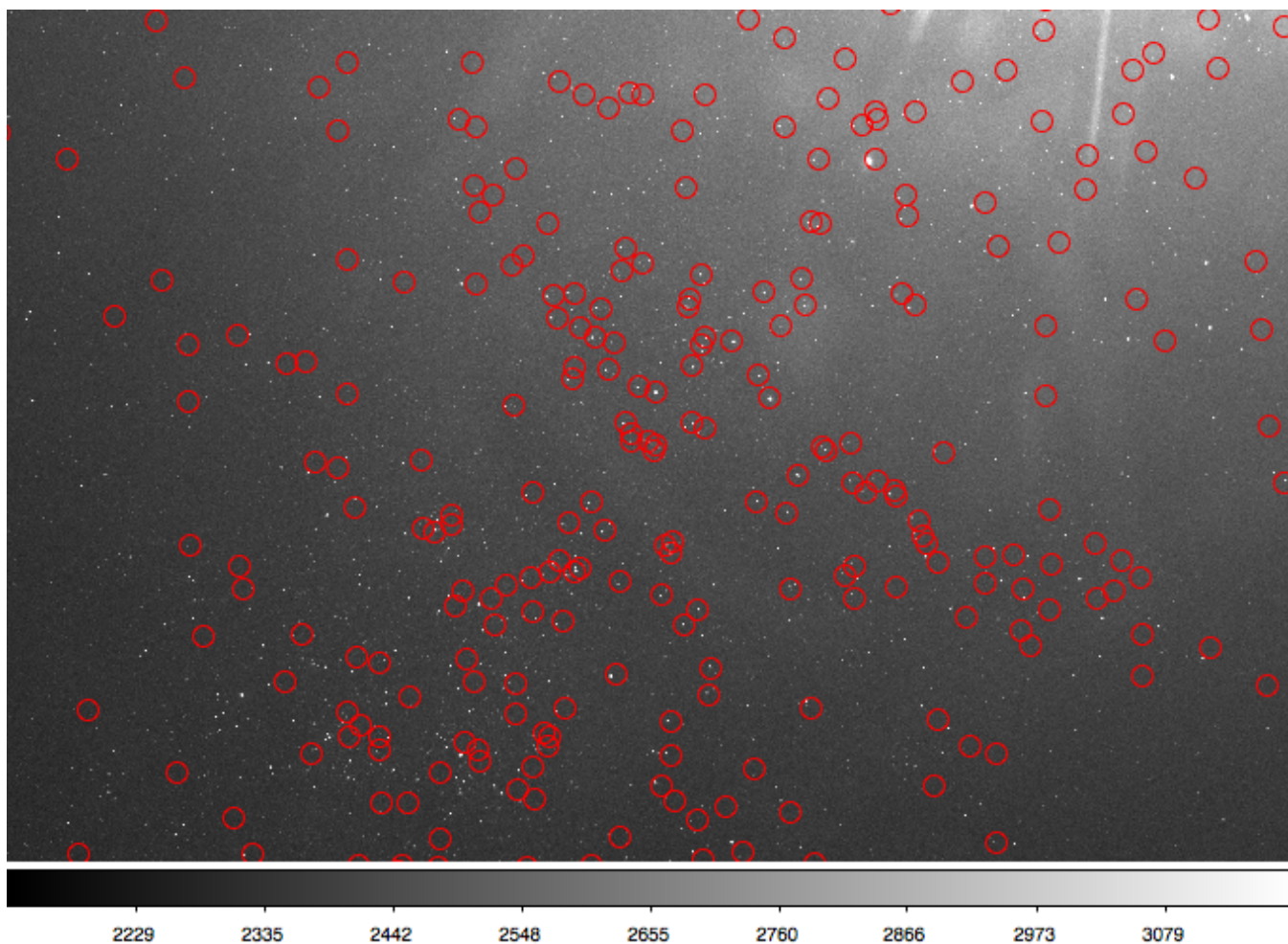
```
588 mv ~/Downloads/new-image.fits ./image.460.wcs.fits
```

```
589 sky2xy image.460.wcs.fits @360.radec.cat
```

```
590 sky2xy image.460.wcs.fits @360.radec.cat | grep -v "off" | awk '{print $5, $6}'
```

```
591 sky2xy image.460.wcs.fits @360.radec.cat | grep -v "off" | awk '{print $5, $6}' > 460.xy.cat
```

The stars do line up in the center, but there is a substantial mismatch away from the image center.



The offsets here are from one hour of sky rotation times the distortion map error, and are large!

So we need a better distortion map, or initial WCS solution. Astrometry.net does OK with finding the center (tangent point) but not so well at generating a global WCS solution.

Tried hacking the header of the image that went through astrometry.net, but didn't manage to get Simbad stars to line up in any sensible way.

So for photometric calibration, let's try something different- namely find a Simbad star that passes right overhead, and find it in the appropriate image. We know the following:

- fits files are N up E left
- Image ut011414.0360 was taken at 03:40 UT on Jan 15, and is at RA=102.345 and dec=-28.57.

```
cp65239:calib cstubbs$ awk 'if (($2<-25)&&($2>-30)&&($3<8)) print $0' SimbadBVR.dat > Simbad.zenith.dat
```

```
cp65239:calib cstubbs$ wc Simbad.zenith.dat
```

```
12 60 588 Simbad.zenith.dat
```

```
cp65239:calib cstubbs$ cat Simbad.zenith.dat
```

```
096.18283193 -28.78011494 6.99 6.374 6.24
```

```
110.31183 -29.29997 6.75 11.2 9.500
```

```
118.45526891 -26.26208578 7.85 6.92 6.48
```

```
118.2697246 -26.4142169 7.45 7.55 7.60
```

```
154.53162512 -28.99199955 5.76 5.53 5.5
```

```
155.98031170 -29.64552930 7.603 6.920 6.534
```

```
168.00496442 -26.13667076 7.815 7.037 6.621
```

```
173.066850 -29.263294 6.19 5.671 5.49
194.12554375 -26.46029020 7.35 6.74 6.8
310.81666184 -29.42391078 7.635 6.960 6.583
228.36944608 -25.30934634 7.168 6.46 6.43
303.82246344 -27.03297557 6.63 5.723 5.98
```

here is this file of bright stars that pass overhead: [Simbad.zenith.dat](#)

RA evolves at 15 degrees per hour, and we take 60 images per hour. So an RA of 118.5 is $(118.5-102.3)/15=1.08$ hours later than frame 380. That's one hour

and 5 minutes later. Using MJD this is 0.0450 days later than image 380, at MJD 56672.161285. So we want 56672.20629. that's image 486

what the heck, try astrometry.net? Nope, failed.

IRAF MSCTPEAK astrometric fit to ut011214.0650.M.long

IRAF MSCTPEAK data base file: [SAO.ut011214.0650.db](#)

SAO Catalog extract covering full image: [SAO.ut011214.0650.cat](#)

Conversion of RA-DEC to Pixel X-Y using the plate solution: [SAO.ut011214.0650.xy.cat](#)

Gemini South:

Latitude: -30:14:26 (-30.2406)

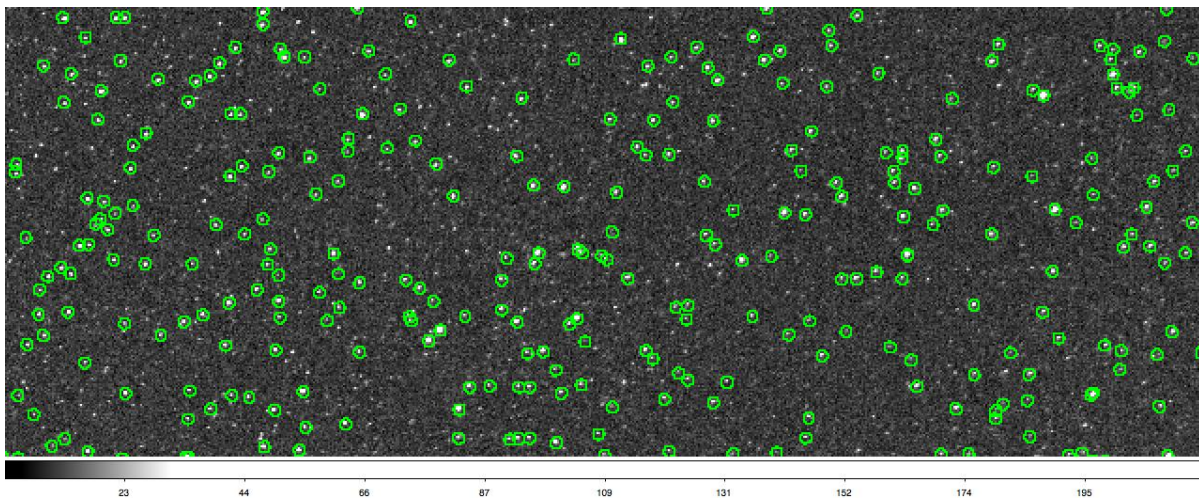
Longitude: -70:44:12

UT Date for exposure: 01-13-14 08:39:39

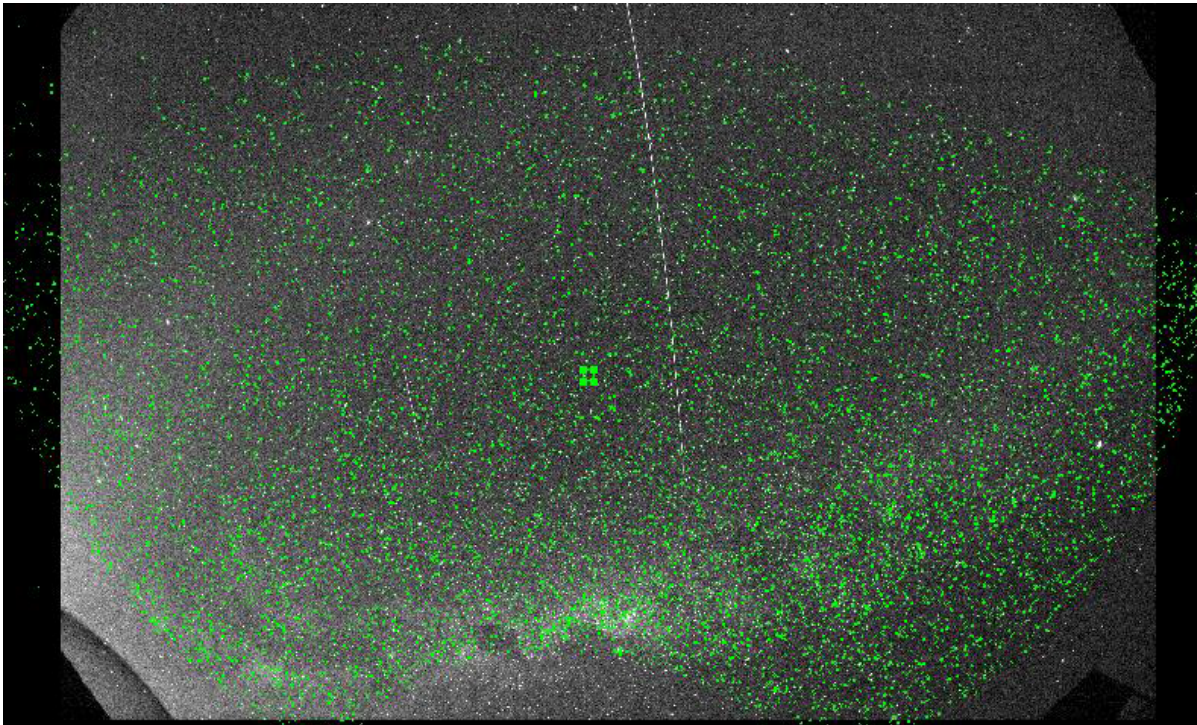
Local Sidereal Time: 11:27:31 HH:MM:SS (170.3792 deg)

Using the above parameters and plate solution the **zenith** pixel coordinates are **x = 1407.48** and **y = 923.20**.

Modest success with this approach. Although, have not yet figured out how to get the fit in the database file (.db) into a full high order WCS fits header.



A small section of ut011214.0650 showing the predicted position of identified SAO stars marked with green circles.



Full image with SAO stars marked along with the zenith point based on the plate solution.

Try a totally different approach. Lens converts alt,az into x,y. Use that as the fundamental representation.

Generated a couple of Matlab functions, one that goes from x,y to alt,az and the other that goes from alt,az to RA,DEC. use this to test thing out.

ut011414.0306.long.M.fits 2047.889 10.400000 56672.129688 67.4674 63.1341 84.2622 38.4578 1073 is a decent image and photometry file. It was taken on Jan 15 UT at 3.11251 hour night, so 03:06:45. That's a time array of time=[2014 01 15 03 06 45].

That didn't work so well.. so let's try going from RA,Dec for Simbad stars to pixel x,y using alt,az as an intermediate coordinate system. Converting Simbad RA Dec to x,y for frame 306 above.

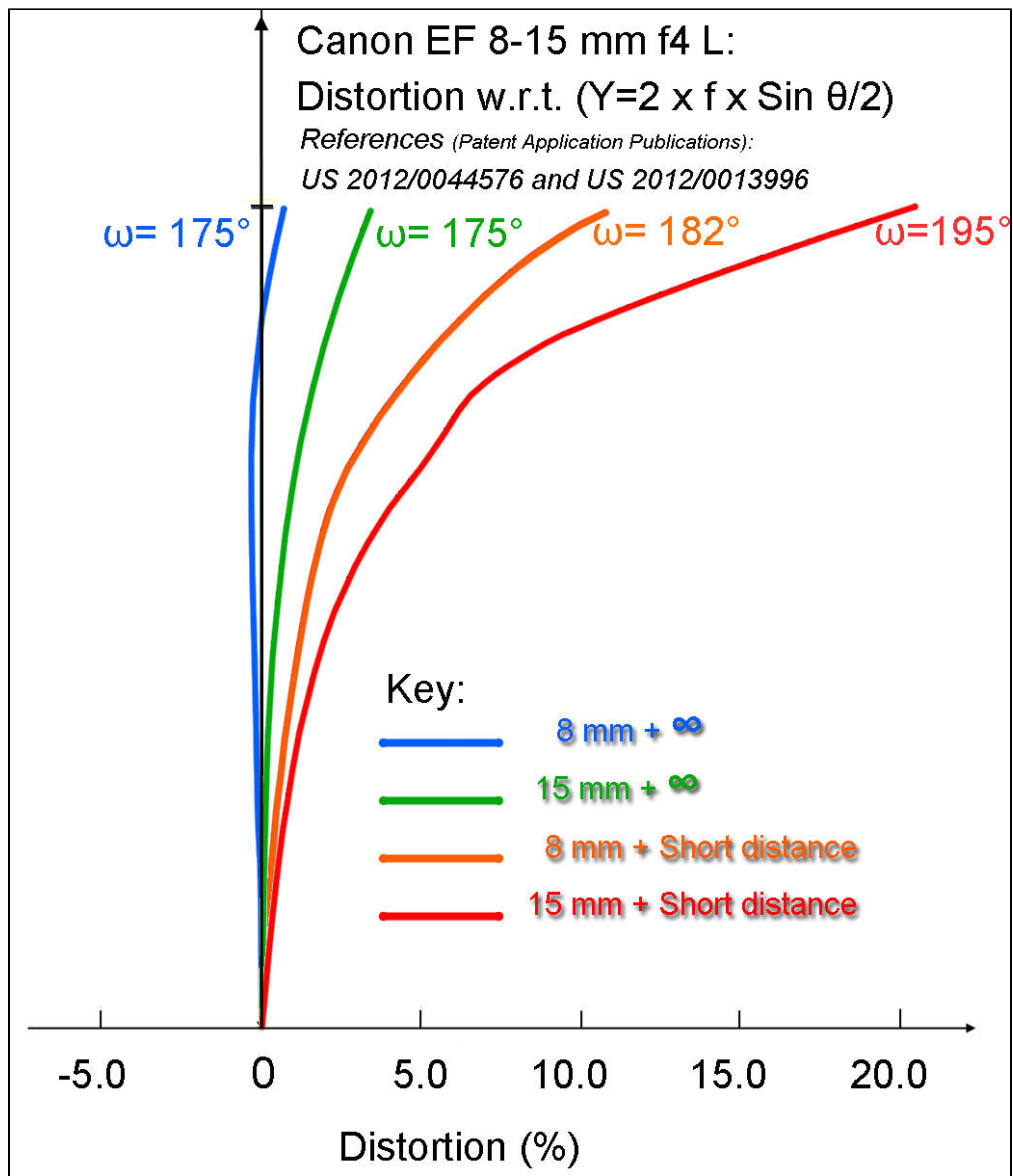
Nope, plots of stellar positions on image 306 don't make much sense.

Jan 17 2014.

One way to debug this positioning code is to get the positions of the sun and the moon right. Using Skycalc for Pachon we get

date	sun RA	sun Dec	sun alt	sun az	moon RA (midnight)	moon Dec (midnight)	moon alt	moon az
Jan 14, noon local	19:44:06	-21:14:30	63.3	77.0	06:56:08 jan 14/15	18:48:36	38.8	18.7
Jan 15, noon local	19:49:24	-21:03:33	63.2	76.7	07:45:42 jan 15/16	16:54:09	36.6	32.2
Jan 16, noon local	19:53:41	-20:52:11	63.0	76.4	08:34:04 jan 16/17	14:18:20	32.4	45.0
Jan 17, noon local	19:57:57	-20:40:25	62.8	76.1	09:21:21 jan 17/18	11:08:27	27.4	56.3

Evidently most of these DSLR fisheye lenses are constructed such that $r = \alpha \sin(\theta/2)$, with $\alpha = 2f$. See http://michel.thoby.free.fr/Canon_8-15mm/8-15mm_review.html which shows



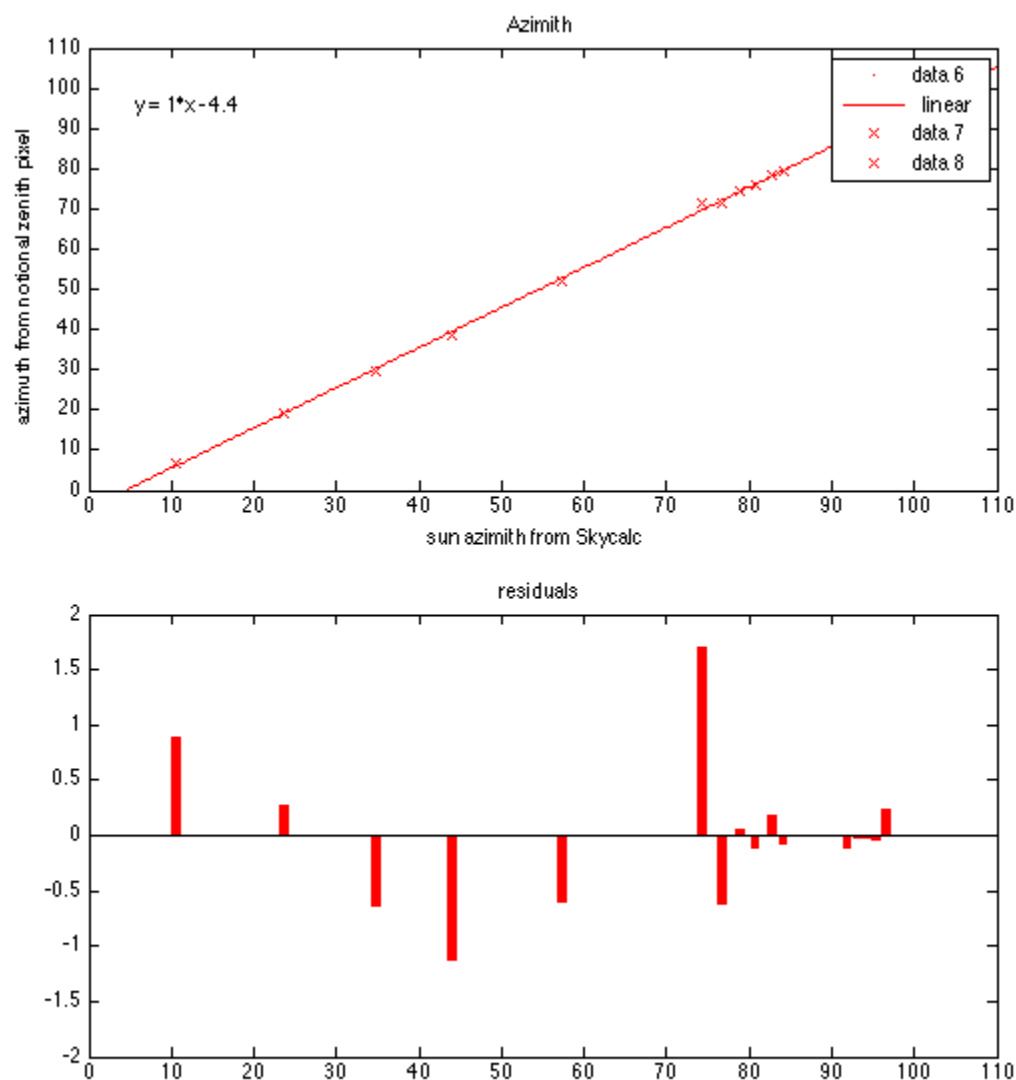
Jan 18 2014.

Took the daycal images from ut011714 and manually determined the x,y position of the center of the sun. Also used skycalc to determine sky coords of sun for times of observations. Results:

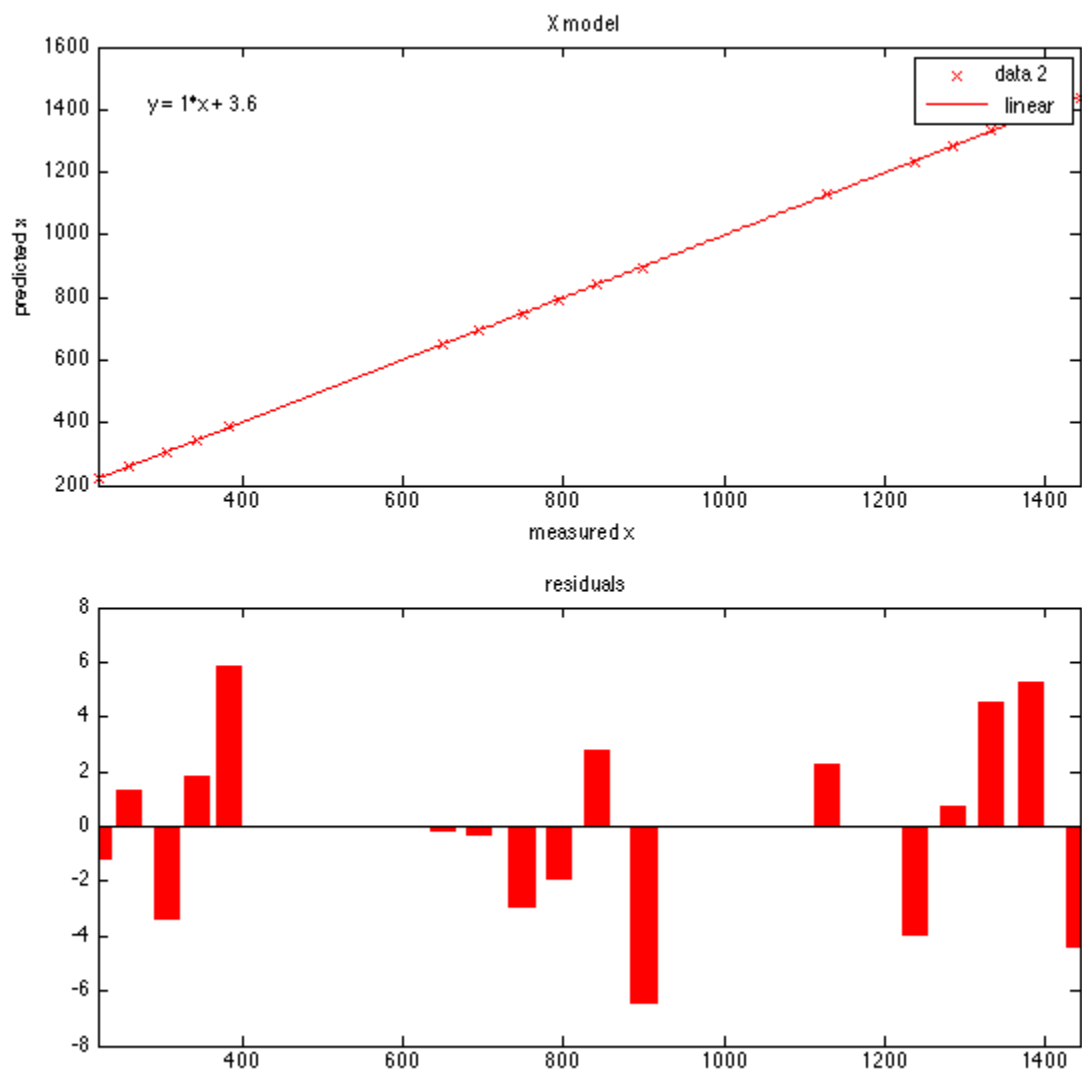
# file	DATE-OBS	MJD-OBS	sun x sun y		sunalt	sunaz	sunRA	sunDec	LMST
ut011714.daycal.0001.fits	2014-01-17T12:39:04	56674.527130	219	881	32.6	96.5	299.38667	-20.69278	
ut011714.daycal.0002.fits	2014-01-17T12:49:12	56674.534167	258	910	34.8	95.4	299.39417	-20.69139	
ut011714.daycal.0003.fits	2014-01-17T12:59:24	56674.541250	305	933	37.0	94.2	299.40167	-20.69000	
ut011714.daycal.0004.fits	2014-01-17T13:09:35	56674.548322	343	955	39.2	93.0	299.40917	-20.68861	
ut011714.daycal.0005.fits	2014-01-17T13:19:47	56674.555405	383	977	41.4	91.8	299.41667	-20.68722	16.40722
ut011714.daycal.0006.fits	2014-01-17T14:16:47	56674.594988	649	1066	53.7	84.1	299.45875	-20.67944	17.35972
ut011714.daycal.0007.fits	2014-01-17T14:26:55	56674.602025	695	1071	55.8	82.8	299.46583	-20.67806	
ut011714.daycal.0008.fits	2014-01-17T14:37:02	56674.609051	748	1089	58.0	80.7	299.47333	-20.67667	
ut011714.daycal.0009.fits	2014-01-17T14:47:10	56674.616088	795	1097	60.1	78.8	299.48083	-20.67528	
ut011714.daycal.0010.fits	2014-01-17T14:57:18	56674.623125	841	1114	62.3	76.7	299.48833	-20.67389	

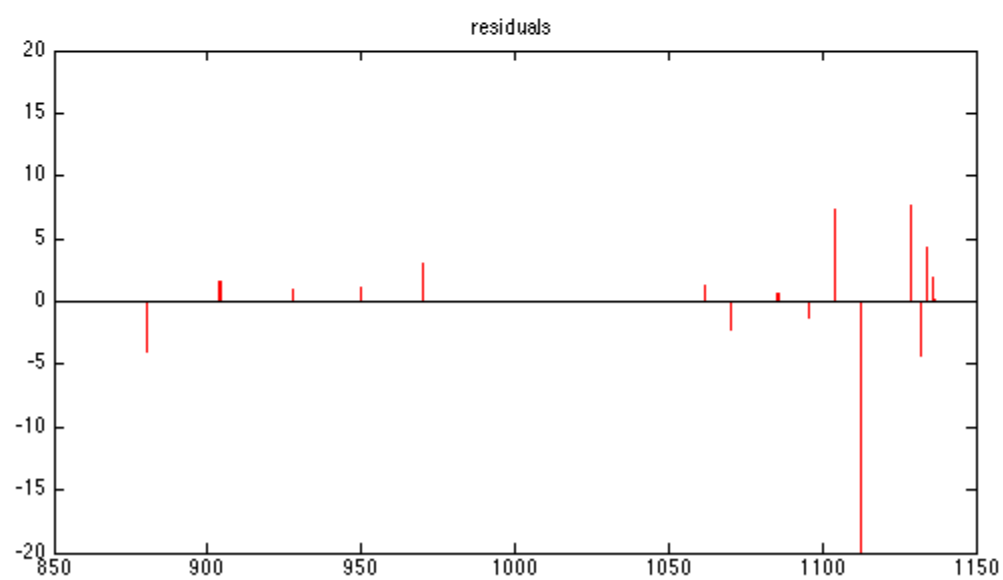
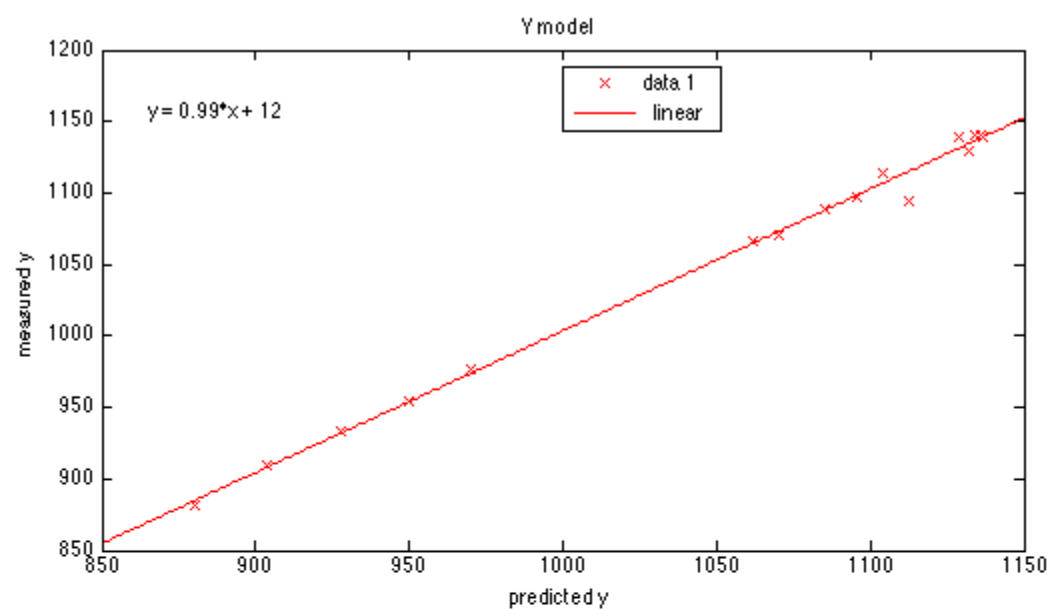
ut011714.daycal.0011.fits	2014-01-17T15:07:26	56674.630162	900	1095	64.4	74.3	299.49583	-20.67250
ut011714.daycal.0012.fits	2014-01-17T15:54:53	56674.663113	1129	1141	73.8	57.3	299.53042	-20.66611
ut011714.daycal.0014.fits	2014-01-17T16:15:07	56674.677164	1238	1139	77.2	44.0	299.54500	-20.66333
ut011714.daycal.0015.fits	2014-01-17T16:25:15	56674.684201	1285	1140	78.6	34.8	299.55250	-20.66194
ut011714.daycal.0016.fits	2014-01-17T16:35:23	56674.691238	1332	1140	79.6	23.6	299.56000	-20.66056
ut011714.daycal.0017.fits	2014-01-17T16:45:31	56674.698275	1383	1130	80.3	10.5	299.56750	-20.65917
ut011714.daycal.0018.fits	2014-01-17T16:55:38	56674.705301	1444	1139	80.4	356.4	299.57458	-20.65778

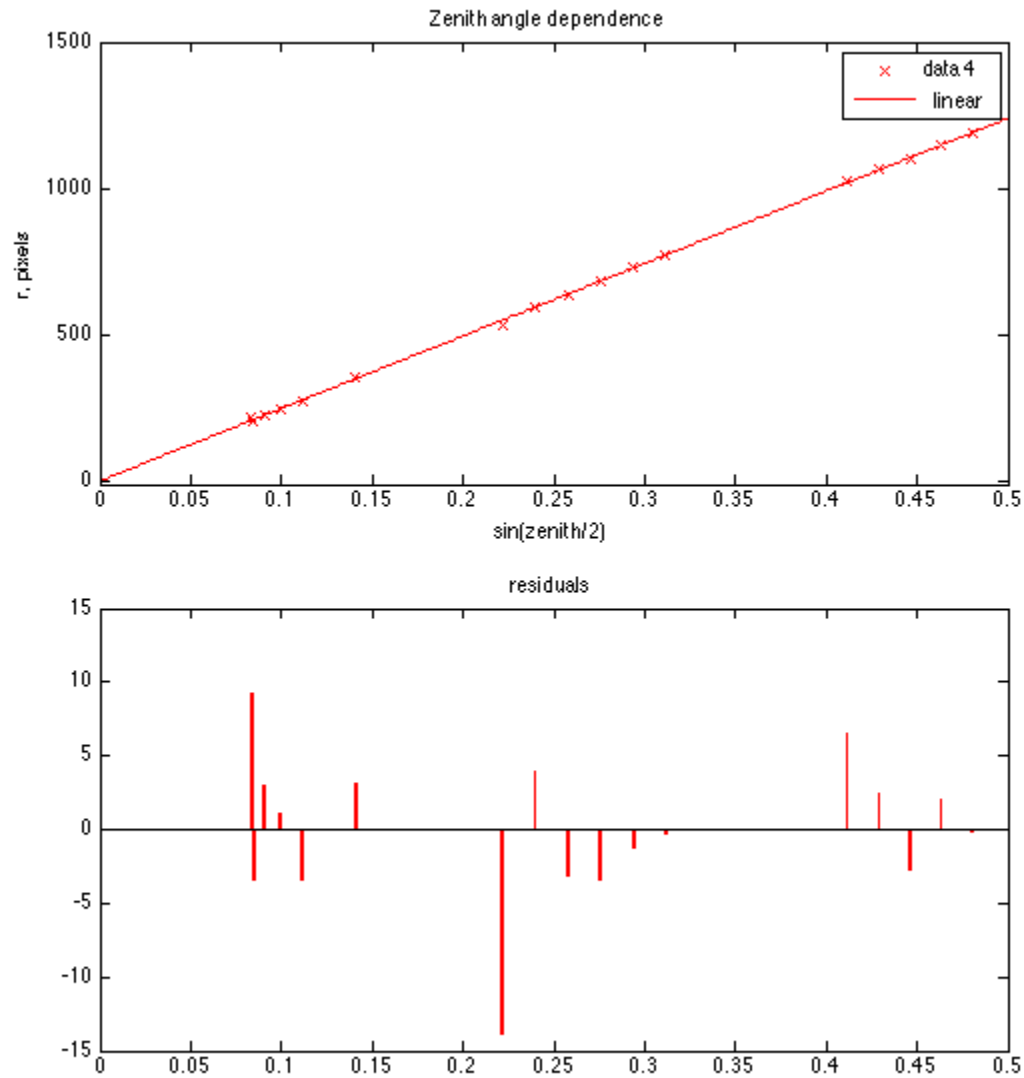
Let's estimate zenith-pointing pixel. Image ut011314.0360.long.fits was taken at UT 01 14 03:39:57. The LMST then was 06 31 11, which is an RA of 97.79583. Zenith is same as our latitude, namely -30.252090. So, running sky2xy on the 360 wcs header, I get 97.79583 -30.252090 J2000 -> 1407.770 923.766. So let's take the zenith pixel to be (1408,924). We can use that information to produce the x',y' location of the sun, with x'=x-1408, y'=y-924.



That worked well! Best-fit line is $r(\text{pixels}) = 2474.4 \cdot \sin(\text{zenith}/2) + 1.6748$. How about in the azimuth direction? Take apparent azimuth as being CCW angle from positive y axis to location, centered on (1408,924). That seems OK as well, with apparent azimuth on chip = $0.99765 \cdot \text{azimuth} - 4.4343$, where apparent azimuth was computed with $\text{app az} = (180/\pi) \cdot (\text{atan2}(-x', y'))$, where the order of the argument and the $\ominus$ sign account for the azimuth convention. This is enough to be able to estimate a source's location on the sensor, given its alt, az.



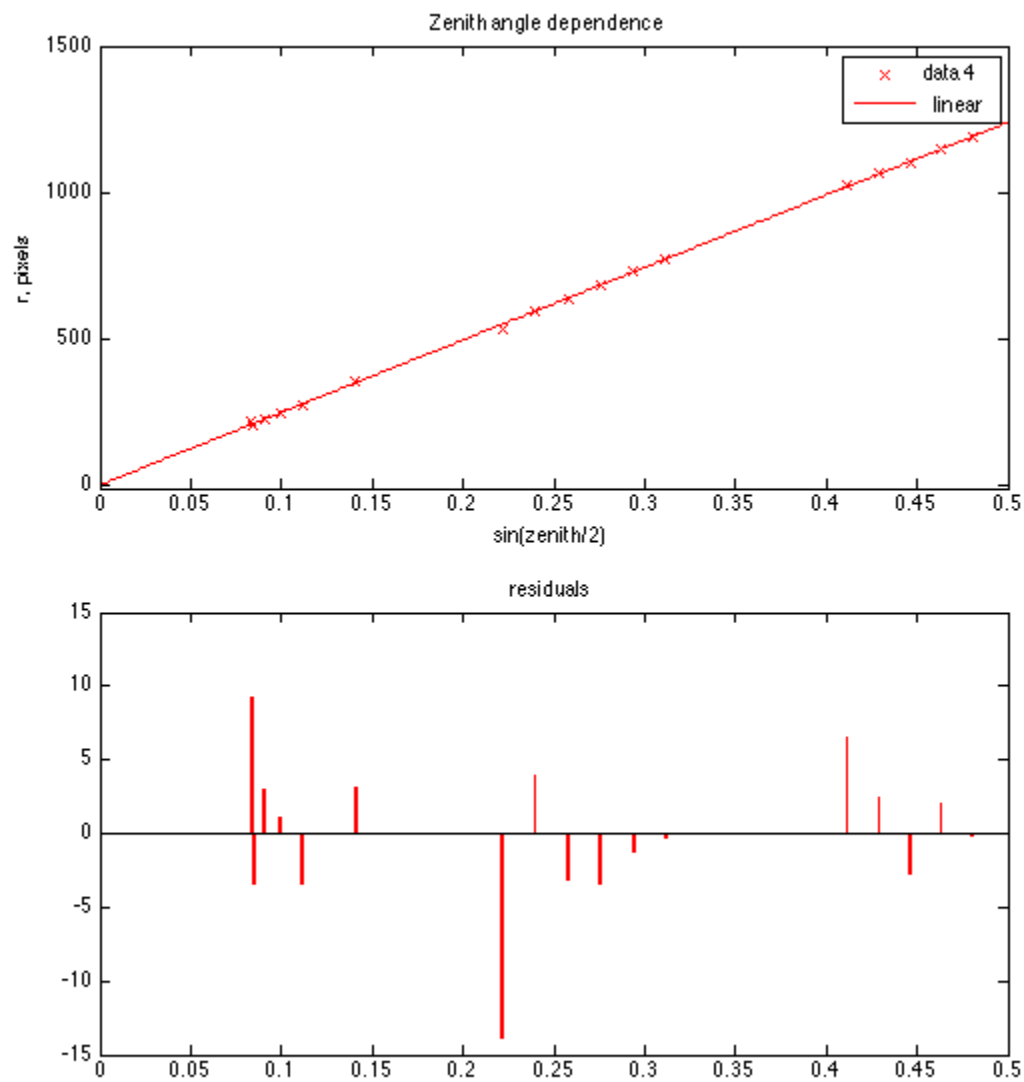




- 1) compute zenith distance with $\text{zenith} = 90 - \text{alt}$
- 2) compute pixel radius off zenith, $\text{pixrad} = 2474.4 * \sin(\text{zenith}/2)$
- 3) compute bearing to that location. $\text{Pixaz} = \text{Azimuth} - 4.4 \text{ deg}$
- 4) compute $\text{dx} = -1 * \sin(\text{Pixaz})$, $\text{dy} = \cos(\text{Pixaz})$
- 5) compute $x = 1408 + \text{dx}$, $y = 924 + \text{dy}$.

Test this with AltAz2xy code in matlab:

[AltAz2xy.m](#)



Verified that I can also go the other way, from xy2AltAz:

[xy2AltAz.m](#)