Lab #3 "Astrometry from CCD Images"

Lab report is due at 4pm on Dec 6 (Monday). Submission on Quercus

Asteroids do not concern me, Admiral. – Darth Vader

1. Overview and Goals

In this lab we will use CCD images to measure the positions of stars relative to the celestial coordinate system and determine proper motions (= the apparent motion of a celestial object) of asteroids. We will use data from the Dunlap Institute telescope (DIT), which was a 50-cm robotic telescope located on Mt. Joy in the Sacramento Mountains of New Mexico, U.S.A. The purpose of this activity is to learn astronomical skills that we will apply to track the position of asteroids (or minor planets). For this lab we will use data that is already posted on the webpage so that we can get off to a quick start and determine the properties of the imaging camera and CCD of DIT. You will learn how to query professional astronomical databases and conduct affine transformation.

Schedule: This is a five-week lab between November 1 and December 6. There will be no class on November 8 during the reading week. Group-led discussions will happen on November 15, 22, and 29. (Check the schedule of Group-led discussion.)

The following sections of this document provide a step-by-step description of: raw astronomical CCD images; CCD calibration and correction; measurement of star positions and comparison to standard star catalogs; and finally, measurement of the position of an asteroid.

2. Key steps

As listed below, there are 10 key steps for this assignment of determining proper motions (= motions of a celestial object on sky) of the asteroid **26-Proserpina**. Before working on the data for the asteroid, however, you need to use data for **NGC 7331** galaxy image in order to learn steps 1–8. And then you can reduce the asteroid data (**26-Proserpina**) with the same steps and extend the analysis with steps 9 and10 to find the associated proper motion of the asteroid.

The key steps are:

1) Download the DIT CCD FITS files. Read and display them using the Python PyFITS library (§3.1);
2) Reduce the data by applying systematic corrections including dark and flat field (§3.3);
3) Compare your image with the digital sky survey and confirm that field is correct (§4);
4) Measure the positions of the stars in your CCD image (§4.1);
5) Cross-correlate the stars in your image with those in the USNO (US Naval Observatory) B-1.0 star catalog (§4.2);
6) Compute "standard coordinates" for the USNO B-1.0 stars and match the two lists (§4.3);
7) Perform a linear least squares fit to find the "plate constants" (§5);
8) Examine the residuals between the measurements and the fit;
9) Use your Python code to compute the position (and associated errors) of the asteroid;
10) Determine your observed asteroid's proper motion (i.e., arcsec per hour) and its measured error, and generate figure(s) that illustrates its observed motion on the plane of the sky.

3. Data

We will be using data obtained with DIT (Dunlap Institute Telescope; Figure 1) which was a 50-cm robotic telescope dedicated to the search for optical transients. The telescope was housed in a dome located at New Mexico Skies on Mt. Joy site near Mayhill, NM, USA. It was equipped with a CCD detector array with 4096 × 4096 pixels, providing a field of view of approximately 36.2 arcminutes × 36.2 arcminutes (or ≈ 0.53 arcsecond per pixel). Details of the facility are listed in Table 1.

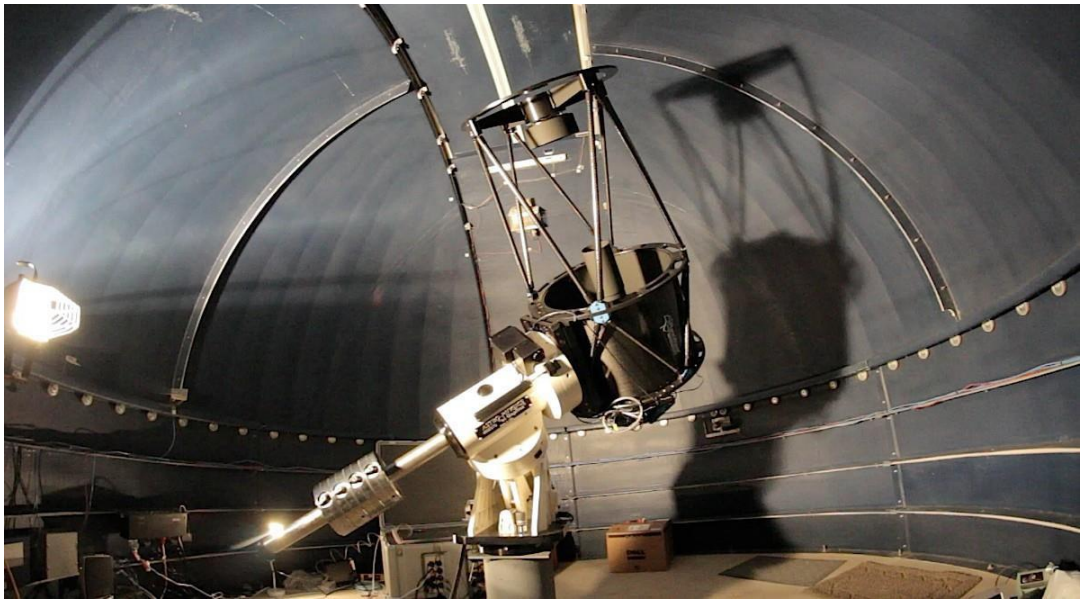

Figure 1: *A picture from inside the dome of the 50-cm DIT telescope on Mt. Joy.*

Table 1: Nominal observatory and instrument properties

Site	Mt. Joy, NM
Geographic location	32°54'10"N 105°31'46"W
Primary mirror	50-cm (diameter)
F/ratio-focal length	F/6.8 – 3454 mm
Camera	Apogee Alta U16M (www.ccd.com)
Detector	TE-cooled Kodak KAF-16803 CCD
CCD format	4096 × 4096, 9.0 µm pixels (unbinned)
Nominal pixel scale	0.53 arc seconds/pixel
Field-of-view	36.2 arc minutes (square)
Optical Filters	*g*,*r*,*i*,*z*,*L*,*B*,clear
Limiting Magnitude	*r* ≈ 17.5/18.5 mag. (10/60 seconds)

CCD FITS files form DIT are available from the following web page

https://drive.google.com/drive/folders/1qaKMqBI1H4NdcqhATHQdzqTF-o6M3kV_?usp=sharing

Note that FITS files contain the image data, stored in binary format, together with some information that records the circumstances under which the data were obtained. **As you already learned, you can use the DS9 package to examine the FITS files.**

3.1 FITS Headers: Although the image data in FITS files are in binary format, each file also has an ASCII preamble that can be conveniently inspected at the Linux command foldby typing, for

example:

[data]$ fold flat006.fits | more

```
SIMPLE  =                    T
BITPIX  =                   16 /8 unsigned int, 16 & 32 int, -32 & -64 real
NAXIS   =                    2 /number of axes
NAXIS1  =                 2048 /fastest changing axis
NAXIS2  =                 2048 /next to fastest changing axis
BSCALE  =    1.0000000000000000 /physical = BZERO + BSCALE*array_value
BZERO   =   32768.000000000000 /physical = BZERO + BSCALE*array_value
DATE-OBS= '2011-10-13T03:18:59' / [ISO 8601] UTC date/time of exposure start
EXPTIME =    2.40000000000E+002 / [sec] Duration of exposure
EXPOSURE=    2.40000000000E+002 / [sec] Duration of exposure SET-
TEMP=  -20.000000000000000 /CCD temperature setpoint in C
CCD-TEMP=  -20.022502500000002 /CCD temperature at start of exposure in C XPIXSZ
=           18.000000000000000 /Pixel Width in microns (after binning) YPIXSZ
=           18.000000000000000 /Pixel Height in microns (after binning)
XBINNING=                    2 / Binning level along the X-axis
YBINNING=                    2 / Binning level along the Y-axis
XORGSUBF=                    0 /Subframe X position in binned pixels
YORGSUBF=                    0 /Subframe Y position in binned pixels
READOUTM= 'Monochrome' /          Readout mode of image
FILTER  = 'r         '           / Filter name
IMAGETYP= 'Light Frame' /         Type of image
JD      =   2455847.6381828706 /Julian Date at start of exposure
FOCALLEN=  0.00000000000000000 /Focal length of telescope in mm APTDIA
=  0.00000000000000000 /Aperture diameter of telescope in mm APTAREA =
0.00000000000000000 /Aperture area of telescope in mm^2
SWCREATE= 'MaxIm DL Version 5.15' /Name of software that created the image
SBSTDVER= 'SBFITSEXT Version 1.0' /Version of SBFITSEXT standard in effect
OBJECT  = 'NGC7331 '             / Target object name
TELESCOP= 'ACP->AstroPhysicsV2' / Telescope name
INSTRUME= 'Apogee USB/Net'       / Detector instrument name
OBSERVER= 'pearl     '           / Observer name
NOTES    = '         '
FLIPSTAT= '          '
CSTRETCH= 'Medium   ' /           Initial display stretch mode
CBLACK  =                 4758 /Initial display black level in ADUs
CWHITE  =                 6798 /Initial display white level in ADUs
PEDESTAL=                    0 /Correction to add for zero-based ADU
SWOWNER = 'Nicholas Law' /     Licensed owner of software
PIERSIDE= 'WEST      '
READMODE= 'Monochrome'
HISTORY File was processed by PinPoint 5.1.8 at 2011-10-13T03:23:09
DATE     = '13/10/11'              / [old format] UTC date of exposure start
TIME-OBS= '03:18:59'            / [old format] UTC time of exposure start UT
= '03:18:59'             / [old format] UTC time of exposure start TIMESYS =
'UTC      '           / Default time system
RADECSYS= 'FK5       '           / Equatorial coordinate system
AIRMASS =    1.01786463149E+000 / Airmass (multiple of zenithal airmass)
ST       = '21 46 26.71'         / Local apparent sidereal time of exp. start
LAT-OBS =    3.29027777778E+001 / [deg +N WGS84] Geodetic latitude
LONG-OBS=  -1.05529444444E+002 / [deg +E WGS84] Geodetic longitude
ALT-OBS =    2.28600000000E+003 / [metres] Altitude above mean sea level
OBSERVAT= 'Dunlap Institute Telescope' / Observatory name
RA       = '22 37 18.00'         / [hms J2000] Target right ascension
OBJCTRA = '22 37 18.00'          / [hms J2000] Target right ascension
DEC      = '+34 26 37.0'         / [dms +N J2000] Target declination
OBJCTDEC= '+34 26 37.0'          / [dms +N J2000] Target declination
CLRBAND = 'R         '           / [J-C std] Std. color band of image or C=Color
HISTORY File was processed by PinPoint 5.1.8 at 2011-10-13T03:23:12
…
…
END
```

The header includes useful information about the size of the image (NAXIS1& NAXIS2), the date (DATE-OBS), the exposure time (EXPTIME), where the celestial coordinates of where telescope was pointing (RA & DEC), and the optical filter in the light path (FILTER). The header is composed of “keywords,” e.g., FILTER with values on the right hand side of the equals sign.

The image data and the keywords are read into Python using the function getdata(), for example to read the file NGC7331-S001-R001-C001-r.fts into the variable xtype

```
In [1]: import pyfits as pf
In [2]: fits1='NGC7331-S001-R001-C001-r.fts'
In [3]: x = pf.getdata(fits1)
In [4]: hdr = pf.getheader(fits1)
In [5]: x.shape
Out[5]: (2048, 2048)
In [6]: x.mean()
Out[6]: 4616.12744140625
In [7]: x.min()
Out[7]: 3473.0
In [8]: x.max()
Out[8]: 65535.0
```

Note that even though the CCD has 4096 × 4096 pixels, the array is only 2048 × 2048 because pixels on the CCD have been "binned," i.e., each 2 × 2 group of pixels in the original array is represented in the `FITS` file by a single value equal to the average of these four

Scrolling through the ASCII FITS header in a terminal is fine for browsing, but if you want to use information in the FITS header in your Python program, you will need to read the keywords. This is accomplished by using the PyFITS function `open`, e.g., to find the object name

```
In [1]: import pyfits as pf
In [2]: fits1='NGC7331-S001-R001-C001-r.fts'
In [3]: hdulist = pf.open(fits1)
In [4]: hdulist.info()
Filename: NGC7331-S001-R001-C001-r.fts
No.    Name         Type      Cards   Dimensions   Format
0    PRIMARY     PrimaryHDU     111  (2048, 2048)  int16
In [5]: hdulist[0].header['object']
Out[5]: 'NGC7331'
In [6]: hdulist[0].data
Out[6]:
array([[ 4136.,  4131.,  4112., ...,  3841.,  3804.,  3971.],
       [ 4098.,  4151.,  4205., ...,  3710.,  3885.,  3934.],
       [ 4134.,  4019.,  3962., ...,  3778.,  3808.,  3932.],
       ...,
       [ 3850.,  3745.,  3840., ...,  3719.,  3817.,  3836.],
       [ 3846.,  3844.,  3785., ...,  3698.,  3742.,  3820.],
       [ 3745.,  3786.,  3759., ...,  3814.,  3665.,  3780.]], dtype=float32)

In [7]: hdulist.close()
```

The function `open` gets the data and the header information. Notice the `close` operation when we are done with the file.

3.2 CCD binning and Overscan: The DIT CCD camera has 4096 × 4096 pixels (see Table 1). As we have already noted in the previous example, the CCD can be operated in a binning mode where blocks of adjacent pixels are combined to yield a "superpixel". Binning is useful because it reduces the size of the resultant FITS file and reduces the effects of read out noise. You can establish whether or not binning is enabled by examining the `XBINNING` and `YBINNING` keywords. For 2×2 binning

```
XBINNING=                    2 / Binning level along the X-axis
YBINNING=                    2 / Binning level along the Y-axis
```

the resultant array is 2048 × 2048 pixels.

Another factor which determines the array size is whether or not the option to digitize the overscan region is enabled. The overscan region comprises 200 columns that are part of the CCD sensor at

the edge of the imaging area, but covered by an opaque mask. The overscan region can therefore be used as a measure of the detector dark. Note that if binning is selected the overscan region cannot be digitized and separate bias and dark exposures must be obtained.

3.3 Bias, Darks, & Flats: At the beginning of each night various calibration frames, including bias, darks, and flats are acquired. The bias frames are zero second exposures that provide a measure of the digital offset when the signal is zero. For dark exposures the shutter does not open; these images can be used to measure the dark current. Flats are twilight sky images–both evening and morning twilight flats are available.

Dark frames are also listed at the start of the log file. The DIT CCD is cooled by a thermoelectric cooler with nominal operating temperature of about 250 K, which ensures that the dark current is negligible. Be sure to check the keyword:

```
CCD-TEMP=   -20.02250250000 /CCD temperature at start of exposure in C
```

to confirm that the cooler is turned on. A histogram of dark current is shown in Figure 2 (what does this plot say about the accuracy of the bias subtraction?). In a 60 s exposure the dark amounts to only 3.2 ADU, which is negligible compared to the sky brightness. When using short exposures (a few minutes or less) and a cooled CCD, dark subtraction can probably be neglected.

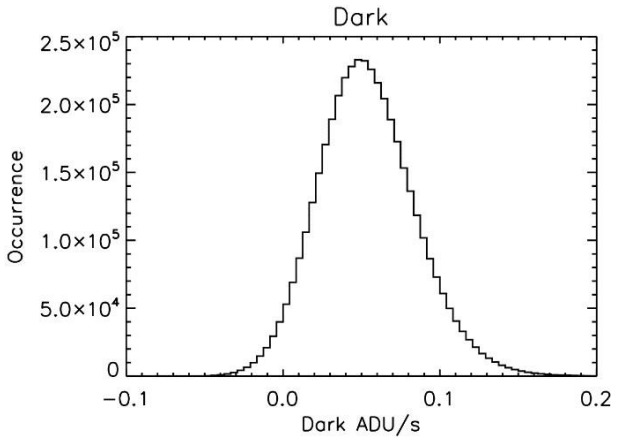

Figure 2: *A dark histogram derived from the average of four bias-subtracted 60 s dark exposures.*

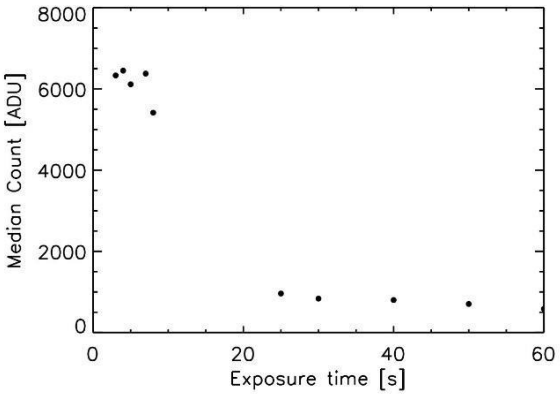

Figure 3: *Median brightness for the bias-subtracted g-band "flat" images taken in twilight. These exposures were taken during twilight. The first image in the sequence is a 3-s exposure, which was taken when the sun was about 5° below the horizon, i.e., just before the start of civil twilight. The long exposures were taken the following morning at when the sun was 8° below the horizon. These images have insufficient counts to make a high signal-to-noise flat.*

The twilight sky provides a convenient source of spatially uniform illumination to measure the pixel-to-pixel response of the camera. The median brightness in *g*-band frames designated as flats are shown in Figure 3. There are two sequences in this example: the short exposures (3–8 s) were taken in evening twilight and have a median level of about 6000 ADU, and a second sequence of frames that were acquired the following morning, which have only a few hundred counts. In this example, only the evening twilight frames are useful. The telescope tracking is turned off for flats, so stars appear to move from one image to the next (see Figure 4).

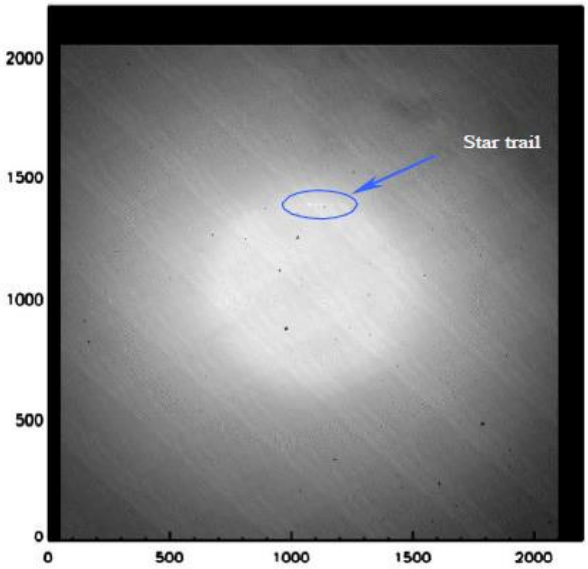

Figure 4: *An average of five evening twilight sky g-band images. The images have been bias subtracted using the bias region on the right. The grey scale runs from 5000 ADU (black) to 7500 ADU (white). Note the overall bulls-eye illumination pattern, which corresponds to 30% variation in response from center to edge. Note also the small-scale diagonal pattern. In addition, there are some local low spots where the response is poor. Near pixel (1120, 1400) the image of a star trail can be seen.*

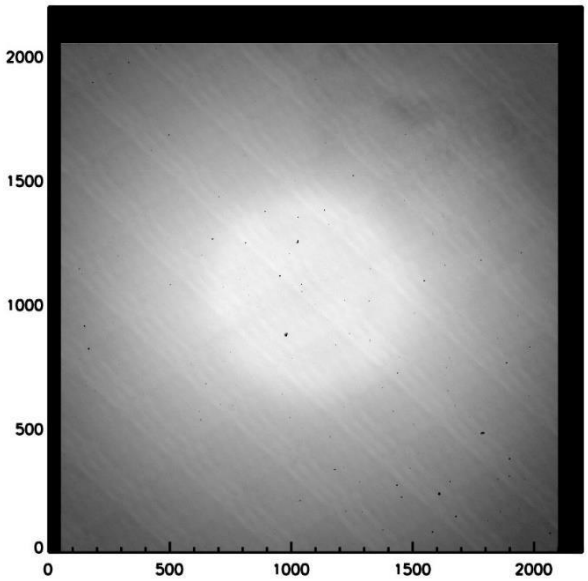

Figure 5: *A g-band flat constructed from the same five frames shown in Figure 4, but combined using a median rather than an average. Each frame was scaled to a median value of 1 before median combining the stack. Note that the star trail visible in Figure 4 has disappeared. The greyscale runs from 0.8 (black) to 1.2 (white).*

4. Star Field

Now that we are ready with a flat-field, we can reduce one of the target fields. In this example we choose a 240-sec *r*-band image of the spiral galaxy NGC 7331, you will need subtract the dark and divide by the appropriate flat.

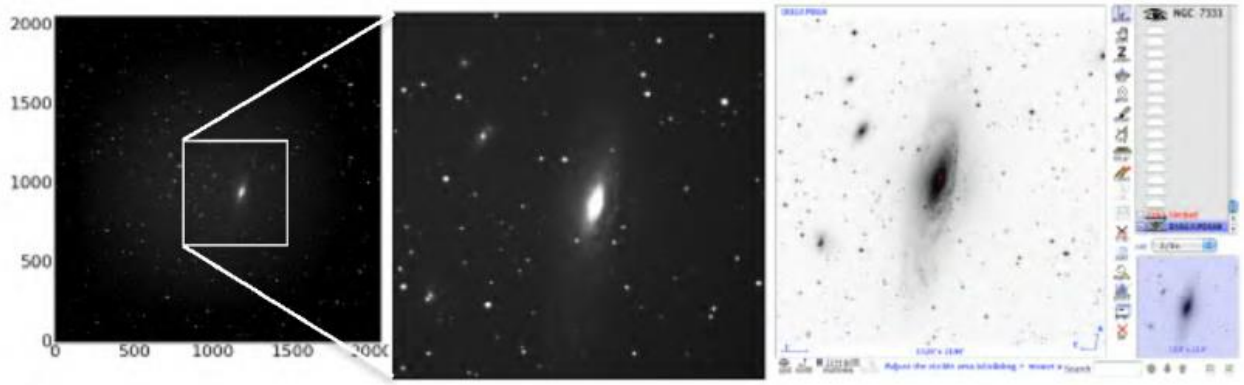

Figure 6: (Left) *A 240-sec r-band image. The image has been flipped up/down so that north is at the top and east to the left.* (Center) *A zoom in from the center of the field.* (Right) *The digital sky survey image for about 13 arcminutes × 14 arcminutes field (RA, DEC) = (22:37:04.102, +34:24:57.3) from ALADIN (see §6 for ALADIN). The default is to use a reverse grey scale so that bright stars show up as the darkest dots.*

Inspection of this NGC 7331 image (Figure 6) reveals several hundred stars and a prominent galaxy near the center of the field. Note that the images have to be flipped up/down using the [::-1] syntax so that images appear in conventional orientation with north at the top and east to the left. To get **imshow** to display the images with pixel (0,0) at the bottom left hand corner you need to edit your **matplotlibrc** file:

```
### IMAGES
#image.aspect : equal                # equal | auto | a number
#image.interpolation  : bilinear     # see help(imshow) for options
image.interpolation  : nearest  # see help(imshow) for options
#image.cmap   : jet                  # gray | jet etc...
image.cmap   : gray                  # gray | jet etc...
#image.lut    : 256                  # the size of the colormap lookup table
image.origin : lower                # lower | upper
#image.resample  : False
```

4.1 Star Positions: Once we have confirmed that the image corresponds approximately to the correct field, the next step is to identify and measure the *x*- and y-positions of the stars in the image. The location of stars is best measured using their centroids

$$\langle x\rangle = \sum_i x_i I_i \Big/ \sum_i I_i , \quad \langle y\rangle = \sum_i y_i I_i \Big/ \sum_i I_i \qquad (1)$$

where the summation i, runs only over a region in the vicinity of the star. Note that the denominator is a measure of the brightness of the star–this is useful information that should be saved along with the centroids.

4.2 Catalogs: We can now compare the observed x- and y-positions on our CCD image with the predicted positions of stars taken from a catalog, such as the USNO-Bl.0 Catalog (see §6 for the use of USNO-B1.0 catalog). The USNO-Bl.0 catalog lists positions, proper motions, and magnitudes in various optical passbands for more than one billion objects over the entire sky. The

data are from scans of photographic plates from the Palomar 48-inch Schmidt telescope taken for various sky surveys during the last century. The USNO-Bl.0 catalog provides all-sky coverage down about V = 21 mag and astrometric accuracy of 0.2 arc second (in J2000 equinox), and should be adequate for our purposes.

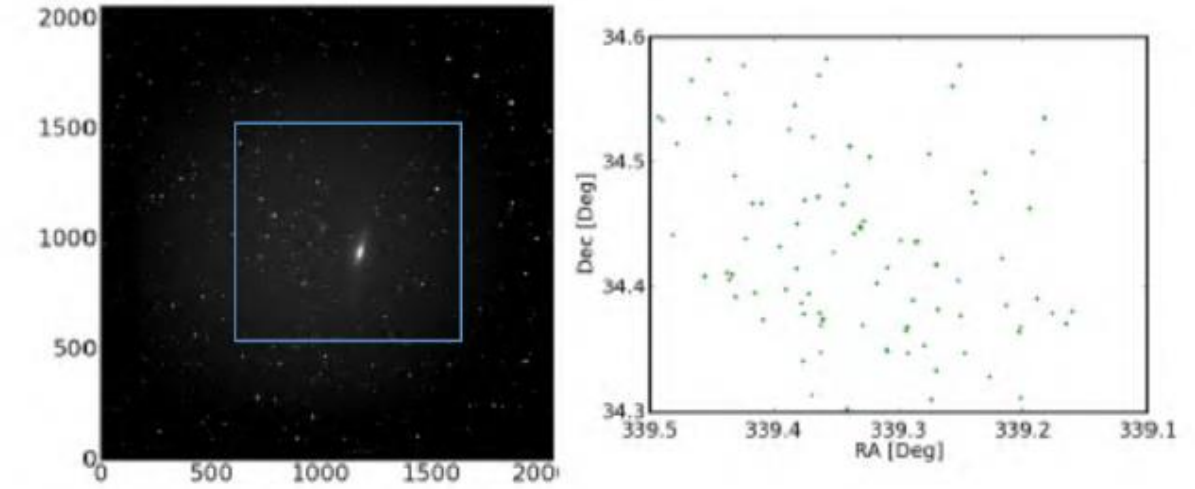

Figure 8: (Left) *The star field in the vicinity of NGC 7331 (same as Figure 6).* (Right) *The positions of stars in a 17 arcminute × 17 arcminute field selected from the USNO-B1.0 catalog (r < 15.0 mag). The blue box represents the 17 arcminutes × 17 arcminutes search field. (Note that plots have equal axes, so why is the blue box square but the plot showing the same region of USNO-B1.0 stars rectangular?)*

You can interrogate the USNO-Bl.0 catalog from within Python (see §9) using the web-based Vizier astronomical catalog database. Below is an example (note: it's a good idea to develop your own program after you understand the example since there could be some changes that you need to make).

```
import pyfits as pf
import matplotlib.pyplot as plt
import numpy as np
import urllib as url

fitsl='NGC7331-S001-R001-C001-r.fts' sl
= f.open(fitsl)
# read position from the FITS file and convert RA/DEC to degrees
ras = s1[0].header['ra']
des = sl[0].header['dec']
radeg = 15*(float(ras[0:2]) + float(ras[3:5])/60. + float(ras[6:])/3600.)
dsgn = np.sign(float(des[0:3]))
dedeg = float(des[0:3]) + dsgn*float(des[4:6])/60. + dsgn*float(des[7:])/3600.
fovam = 17.0 #size of square search field in arc min
name,rad,ded,rmag = usno(radeg,dedeg,fovam)
w = np.where(rmag < 15.)[0]      # select only bright stars r < 15 mag.
plt.plot(rad[w],ded[w],'g.')
plt.locator_params(axis='x',nbins=4)
plt.locator_params(axis='y',nbins=4)
plt.tick_params('x',pad=lO)
plt.xlabel('RA [Deg]')
plt.ylabel('Dec [Deg]')
plt.ticklabel_format(useOffset=False)
plt.axis('scaled')
plt.xlim([339.5,339.1])      #reverse the x-axis direction
```

(Or, you can query the catalog online directly.) Note that by convention right ascension increases to the left. The resultant plot for this example is shown in Figure 8. Inspection of Figure 8 does not

immediately give the impression that we are looking at the same star field. Look at the blue box (17' x 17') and convince yourself that you can trace the same pattern of stars. To make sure you have the right stars you can interrogate the USNO B-1.0 catalog directly from ALADIN, using the open/surveys menu choice. Clicking on a star in the ALADIN window will show you the catalog entry.

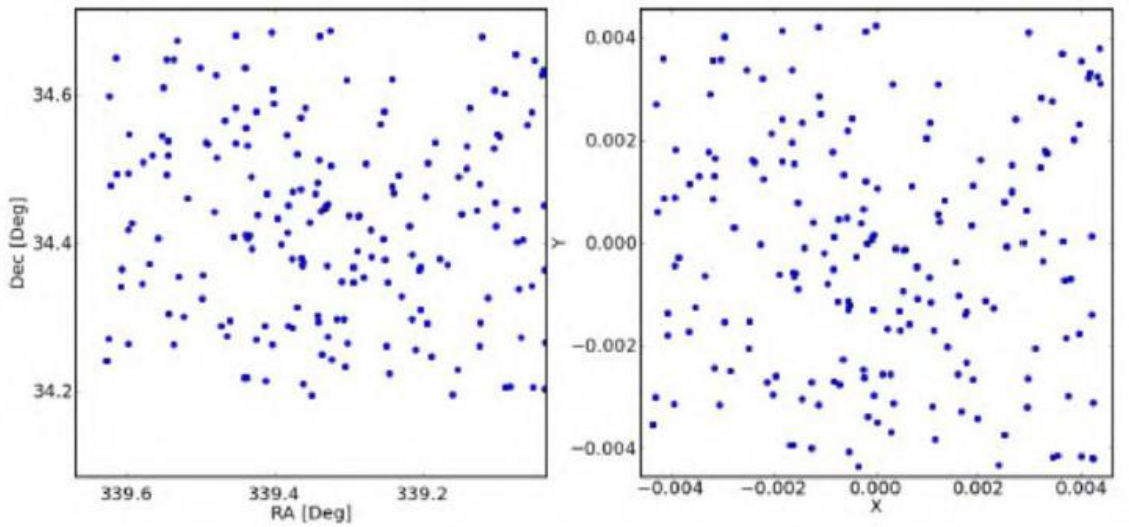

Figure 9: (Left) *The catalog positions of 199 USNO B-1.0 stars within a 30 × 30 arcminute square field centered at the position of NGC 7331 (RA = 339°.325; DEC= +34°.444 2000.0) with r magnitude brighter than 14.5 mag.* (Right) *The same data plotted in standard coordinates (see Eq. 2)—the projection from celestial coordinates on the unit sphere tangent plane.*

4.3 Matching Stars & Standard Coordinates: We now have the positions of the observed stars on the CCD in pixel *x* and y and in right ascension and declination from the USNO B-1.0 catalog. The USNO B-1.0 coordinates are on the celestial sphere, whereas the stars in our CCD image are measured on a plane, which is a projection of the celestial sphere. The first step in the comparison is to project the USNO B-1.0 coordinates from the celestial sphere to the plane represented by the CCD so that we can make a one-to-one comparison of these quantities.

The transformation can be obtained by considering the geometry in Figure 10 (see the detailed derivation in §7). The projected coordinates (*X*, *Y*) of a position on the celestial sphere (α, δ) in the vicinity of a field centered at (α_0, δ_0) are

$$X = -\frac{\cos\delta \sin(\alpha - \alpha_0)}{\cos\delta_0 \cos\delta \cos(\alpha - \alpha_0) + \sin\delta \sin\delta_0}$$
$$Y = -\frac{\sin\delta_0 \cos\delta \cos(\alpha - \alpha_0) - \cos\delta_0 \sin\delta}{\cos\delta_0 \cos\delta \cos(\alpha - \alpha_0) + \sin\delta \sin\delta_0}, \quad (2)$$

where the celestial is taken to be the unit radius.

Conversion from (*X*, *Y*) coordinates to pixel coordinates (*x*, *y*), is straightforward. If *f* is the focal length of the camera and *p* is the pixel size, then for an ***ideal*** camera

$$x = f(X/p) + x_0$$
$$y = f(Y/p) + y_0,$$

where, by ideal we mean that the focal length of the imaging system is constant over the field, there is no anamorphic magnification ($f_x = f_y$), or equivalently the pixels are square ($p_x = p_y$), and that the CCD is oriented so that the *X*-axis lies along the declination small circle. In general, none of these

conditions are true. Optical systems have variable magnification and distortion, pixels may be parallelograms, and the CCD is subject to an unknown rotation with respect to celestial coordinates. However, for a first guess let's adopt the nominal values from Table 1: f = 3454 mm and p = 0.009 mm. The results of this comparison are shown in Figure 11.

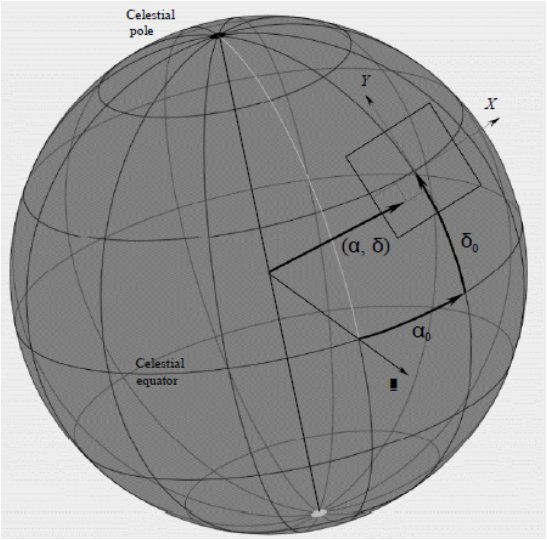

Figure 10. *Projection from celestial coordinates of a point (α, δ) on the sky in a field centered at (α_0, δ_0) onto the tangent plane of a CCD with a position (X, Y). The X-axis is aligned along the small declination circle and the Y-axis is aligned along the great hour circle, and points towards the celestial pole. The filled black symbol indicates the direction of the vernal equinox, which defines the origin of right ascension.*

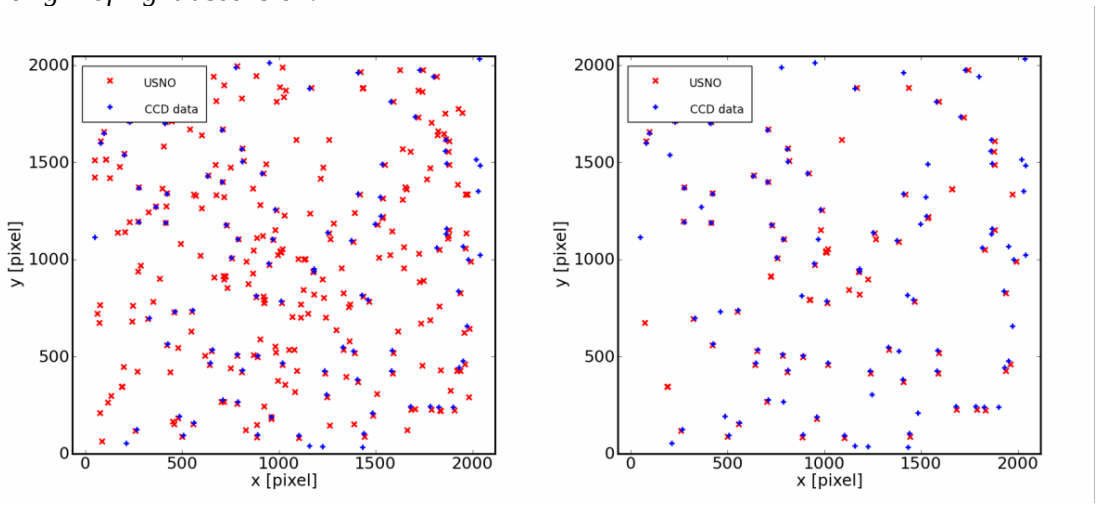

Figure 11: (Left) *Comparison of the measured pixel positions (+) of stars and converted standard positions from the USNO B-1.0 catalog (x) using f = 3454 mm, p = 0.018 mm (x2 binning), and x_0 = y_0 = 1024. The situation is obviously complicated, but many of the CCD stars appear to have counterparts in the USNO B-1.0 catalog.* (Right) *Plotting only the brighter stars (r < 13 mag.) reveals a much clearer picture.*

Evidently, if we plot all the stars (left) then it is difficult to figure out how the star patterns match up; however, if we choose only some of the brighter stars from the USNO B-1.0 list (r < 13 mag.) it is clear that that we see the same pattern with a small offset. Inspection of the stars also suggests that the magnification is not quite right and that two patterns are rotated with respect to one another. By using the distance between the predicted and measured positions it is possible to uniquely match up the stars (Figure 12).

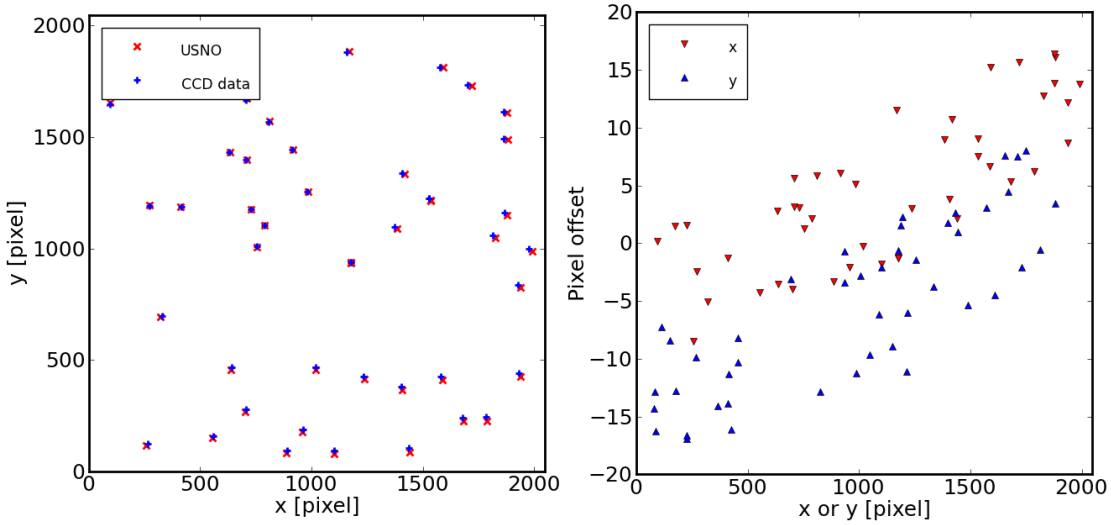

Figure 12: (Left) *Stars from the NGC 7331 field from Figure 11 matched using a distance criterion.* (Right) *The pixel offset between the observed and predicted positions of USNO B-1.0 stars versus or x or y pixel.*

5. Determining the plate constants using least squares

The standard coordinates are defined for the tangent plane to the unit sphere. As we have seen in §4.3 the positions on the CCD depend on the focal length and the pixel size. In general, the CCD is not perfectly aligned so there is a rotation between the (*x*, *y)* CCD-based coordinate system and the (*X*, *Y)* standard coordinates. Taking this into account

$$x = \frac{f}{p}\left(X\cos\theta - Y\sin\theta\right) + x_0$$
$$y = \frac{f}{p}\left(X\sin\theta + Y\cos\theta\right) + y_0 \quad ,$$

where θ is the rotation between the two frames. The general affine transformation between two vector spaces, **X** and **x**, is described by a linear transformation (magnification, shear, rotation) plus a translation (see §8). This transformation is conveniently implemented by matrix multiplication using homogeneous coordinates, where **X**= (*X*, *Y*, 1), **x** = (*x*, *y*, 1) and

$$\mathbf{x} = \mathbf{T}\mathbf{X} \tag{3}$$

where

$$\mathbf{T} = \begin{pmatrix} (f/p)a_{11} & (f/p)a_{12} & x_0 \\ (f/p)a_{21} & (f/p)a_{22} & y_0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{4}$$

Thus, in the linear approximation there are six unknown "plate constants," which we can solve for using the method of linear least squares. **The constants a_{ij} refer to the scale, shear, and orientation of the image, and x_0 and y_0 are pointing offsets in pixels.**

To find the least-squares solution we write down the equations of condition describing the relation between the independent variables (X_i, Y_i) and our *N* position measurements of the dependent variable from the CCD image, (x_i, y_i), as

$$\begin{aligned} x_1 &= (f/p)a_{11}X_1 + (f/p)a_{12}Y_1 + x_0 \\ x_2 &= (f/p)a_{11}X_2 + (f/p)a_{12}Y_2 + x_0 \\ \cdots &= \cdots \\ x_N &= (f/p)a_{11}X_N + (f/p)a_{12}Y_N + x_0, \end{aligned} \tag{5}$$

and

$$\begin{aligned} y_1 &= (f/p)a_{21}X_1 + (f/p)a_{22}Y_1 + y_0 \\ y_2 &= (f/p)a_{21}X_2 + (f/p)a_{22}Y_2 + y_0 \\ \cdots &= \cdots \\ y_N &= (f/p)a_{21}X_N + (f/p)a_{22}Y_N + y_0. \end{aligned} \tag{6}$$

Or in compact matrix form for Eq. 5 and Eq. 6 for can be represented as

$$\mathbf{a} = \mathbf{Bc}, \tag{7}$$

where for x we have

$$\mathbf{a} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} (f/p)X_1 & (f/p)Y_1 & 1 \\ (f/p)X_2 & (f/p)Y_2 & 1 \\ \vdots & \vdots & \vdots \\ (f/p)X_N & (f/p)Y_N & 1 \end{pmatrix}, \text{ and } \mathbf{c} = \begin{pmatrix} a_{11} & a_{12} & x_0 \end{pmatrix},$$

with a similar equation for y.

It is easy to compute χ^2 in matrix form using Eq. 7

$$\chi^2 = (\mathbf{a} - \mathbf{Bc})^T (\mathbf{a} - \mathbf{Bc}) \tag{8}$$

As we have seen in class, we can solve for **c** from by multiplying each side Eq. 7 by the transpose of **B**, $\mathbf{B}^T$, followed by the inverse matrix $(\mathbf{B}^T\mathbf{B})^{-1}$, i.e.,

$$\mathbf{c} = \left(\mathbf{B}^T\mathbf{B}\right)^{-1}\mathbf{B}^T\mathbf{a}\,. \tag{9}$$

A similar expression is used for y.

The results of applying Eq. 9 to find the matrix elements **c** are summarized in Figure 13. In the initial match most stars show very small residuals, suggesting that the match is reliable except perhaps for three stars that may be mismatched. When these are removed the overall quality of the fit is improved.

The coefficients of the solution can be determined assuming the nominal values f/p =191900 pixels per radian. Recall that the geometric interpretation of the determinant of a 2-d transformation is the scale factor for area (see Eq. 21). Therefore, a convenient measure of the effective telescope scale f/p (in pixels per radians) is

$$f/p = \sqrt{\det(\mathbf{T})} = 190020 \text{ pixels/radian}, \quad (10)$$

where **T** is the matrix in Eq. 3 and the matrix elements of **T** are found using Eq. 9. Assuming that the pixel size is known with precision (p = 0.018 for 2-pixel binning) then the effective focal length is 3420 mm, or about 1% shorter than the nominal value. You should inspect the matrix element coefficients and determine the skew-ness of the pixel size (is it better than 1 part in 100 or 1,000?). What is the rotation amount between the sky (true north) and CCD columns (see Eq.19)?

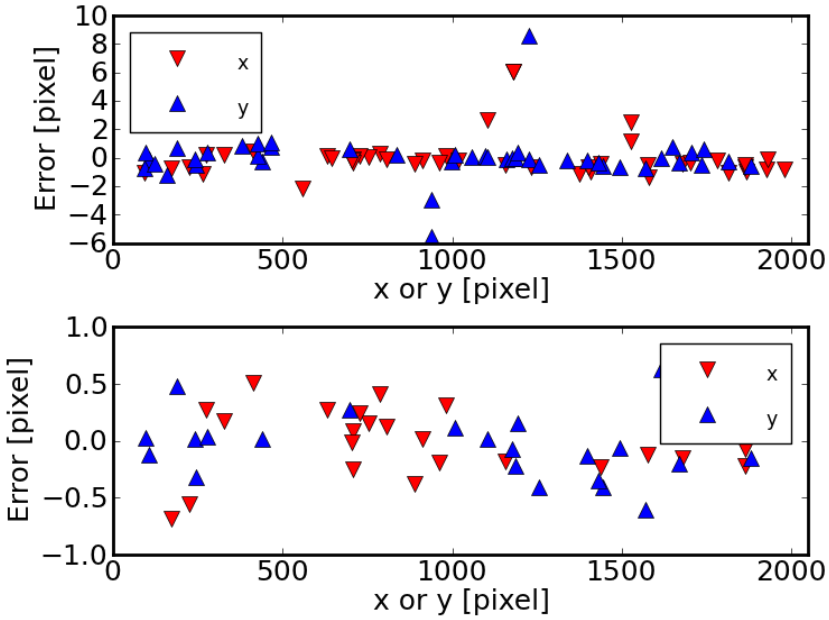

Figure 13. (Top) *Errors in x and y relative to a linear least squares solution. In the top plot the all stars are matched between the CCD and the USNO B-1.0 catalog. Of these matches five have suspiciously large residuals.* (Bottom) *If we remove matches with residuals larger than 2 pixels then the bottom plot results. The stars with poor matches (residuals greater than one pixel) may be false matches. The rms x and y errors are 0.48 pixels and 0.40 pixels respectively.*

The least-squares solution allows us to compute the standard coordinates for any pixel in the CCD. To convert from a measured (x, y) CCD position to (α, δ) in celestial coordinate we first convert to standard coordinates, using the inverse of Eq. 3, i.e.,

$$\mathbf{X} = \mathbf{T}^{-1}\mathbf{x} \, . \quad (11)$$

The conversion from standard coordinates to celestial coordinates is then given by Eq. 16 (see §7). Your group will need to follow the same exercise outlined above to derive the plate constants for the CCD camera.

6. Online Resources

The key online resource for this assignment is the **USNO B-1.0** catalog from which you can retrieve the coordinates and brightness (= magnitudes) of stars around your targets: NGC 7331 and 26-Proserpina. The catalog is available at

https://vizier.u-strasbg.fr/viz-bin/VizieR-3

Note that you can access the catalog in your Python code as explained above and §9. More information about the USNO B-1.0 catalog can be found at

http://tdc-www.harvard.edu/software/catalogs/ub1.html

ALADIN is a simple and excellent tool to see images of any part of sky available at

https://aladin.u-strasbg.fr/aladin.gml

You can download ALADIN Desktop on your computer and check images around your targets. In addition, you can overlap the USNO B-1.0 catalog on the image. For example, type "NGC 7331" for Command, choose SDSS, J2000 (for Frame), and File-Catalog(VizieR)-I-Astrometric Data-USNO-B1.0 Catalog-Load.

The "**What's Observable Tool**" from JPL provides the list of asteroids and comets that are observable on a certain date given a set of constraints on the observation conditions, the object type, and its orbital and physical properties. It's available at

https://ssd.jpl.nasa.gov/tools/sbwobs.html#/

7. Appendix: Tangent plane projection

Consider the Cartesian coordinate system (x, y, z) with the x-axis pointed towards some right ascension α_0, and the x-y plane coincident with the celestial equator (Figure 14 for illustration).

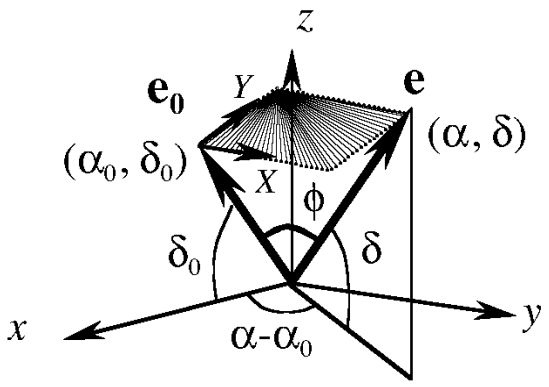

Figure 14. *A coordinate system with the x-axis pointed towards some right ascension α_0, and the x-y plane coincident with the celestial equator. The angle between the directions (α, δ) and (α_0, δ_0) is φ. The tangent plane is normal to the unit vector e_0. The tangent plane coordinates are (X, Y).*

The components of the unit vector **e** pointing towards a star at (α, δ) are

$$\begin{aligned} x &= \cos\delta\cos(\alpha-\alpha_0) \\ y &= \cos\delta\sin(\alpha-\alpha_0)\,. \\ z &= \sin\delta \end{aligned} \tag{12}$$

The angle between the directions $\mathbf{e}(\alpha, \delta)$ and $\mathbf{e}_0(\alpha_0, \delta_0)$ ϕ is given by computing the dot product

$$\begin{aligned} \cos\phi &= \mathbf{e}_0 \cdot \mathbf{e} \\ &= \cos\delta\cos(\alpha-\alpha_0)\cos\delta_0 + \sin\delta_0\sin\delta. \end{aligned} \tag{13}$$

Consider the plane normal to $\mathbf{e}_0$ with X oriented along the y-axis and Y rotated about y by an angle equal to δ_0 such that the unit vectors in this plane are

$$\hat{\mathbf{X}} = (0 \quad 1 \quad 0); \quad \hat{\mathbf{Y}} = (\sin\delta_0 \quad 0 \quad -\cos\delta_0).$$

A point in the tangent plane $\mathbf{p}$ is

$$\mathbf{p} = \mathbf{e}_0 - \left(X\hat{\mathbf{X}} + Y\hat{\mathbf{Y}}\right) = p\mathbf{e}\,,$$

where $p = 1/\cos\phi = (1+X^2+Y^2)^{1/2}$.

When written in component form we have three simultaneous equations relating (α, δ) and (X, Y)

$$\begin{pmatrix} \cos\delta_0 - Y\sin\delta_0 \\ -X \\ \sin\delta_0 - Y\cos\delta_0 \end{pmatrix} = \begin{pmatrix} \dfrac{\cos(\alpha-\alpha_0)\cos\delta}{\cos(\alpha-\alpha_0)\cos\delta\cos\delta_0 + \sin\delta\sin\delta_0} \\ \dfrac{\sin(\alpha-\alpha_0)\cos\delta}{\cos(\alpha-\alpha_0)\cos\delta\cos\delta_0 + \sin\delta\sin\delta_0} \\ \dfrac{\sin\delta}{\cos(\alpha-\alpha_0)\cos\delta\cos\delta_0 + \sin\delta\sin\delta_0} \end{pmatrix}, \tag{14}$$

which yield

$$\begin{aligned} X &= -\frac{\sin(\alpha-\alpha_0)\cos\delta}{\cos(\alpha-\alpha_0)\cos\delta\cos\delta_0 + \sin\delta\sin\delta_0} \\ Y &= -\frac{\cos(\alpha-\alpha_0)\cos\delta\sin\delta_0 - \sin\delta\cos\delta_0}{\cos(\alpha-\alpha_0)\cos\delta\cos\delta_0 + \sin\delta\sin\delta_0} \end{aligned}\,. \tag{15}$$

Expressions for α and δ as a function of X and Y can be found from

$$\begin{aligned} \tan(\alpha-\alpha_0) &= -\frac{X}{\cos\delta_0 - Y\sin\delta_0} \\ \sin\delta &= \frac{\sin\delta_0 + Y\cos\delta_0}{\left(1+X^2+Y^2\right)^{1/2}} \end{aligned}\,, \tag{16}$$

by using inverse trig functions.

8. Appendix: Affine Transformation

The affine transformation described by Eq. 3 comprise

$$\mathbf{M} = \begin{pmatrix} m_x & 0 & 0 \\ 0 & m_y & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{18}$$

where m_x and m_y are the magnification in the x- and y-directions, rotation,

$$\mathbf{R} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{19}$$

where θ is a counterclockwise rotation, and shear ,

$$\mathbf{S}_h = \begin{pmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \text{ and } \mathbf{S}_v = \begin{pmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{20}$$

where s_h is the horizontal shear (x-displacement proportional to y-position.) and s_v is the vertical shear. The product of these matrices (magnification, rotation, shear, & translation) yield:

$$\mathbf{TS_hS_vRM} = \begin{pmatrix} m_x\left[\left(1 + s_h s_v\right)\cos\theta + s_h\sin\theta\right] & m_y\left[s_h\cos\theta - \left(1 + s_h s_v\right)\sin\theta\right] & \Delta x \\ m_x\left(s_v\cos\theta + \sin\theta\right) & m_y\left(\cos\theta - s_v\sin\theta\right) & \Delta y \\ 0 & 0 & 1 \end{pmatrix}$$

and the determinant is

$$\det\left(\mathbf{TS_hS_vRM}\right) = m_x m_y, \tag{21}$$

and thus the determinant represents the scale factor by which areas are transformed.

9. Appendix: Querying the USNO-B1.0 Catalog

The following Python program will return the star names, the celestial coordinates (RA and DEC in decimal degrees; equinox J2000.0) and the *r*-band magnitude given a search location (also decimal degrees; J2000.0) and the search field (a square region in arc minutes). This routine, which is written for Python v3.0 environment, queries the web-based Vizier astronomical catalog system.

```
#function: query Vizier USNO-B1.0 catalog
def usno(radeg,decdeg,fovam): # RA/Dec in decimal degrees/J2000.0 FOV in arc min.

    import numpy as np
    import urllib.request as url

    #url format for USNO
    str1 = 'http://webviz.u-strasbg.fr/viz-bin/asu-tsv/?-source=USNO-B1'
    str2 = '&-c.ra={:4.6f}&-c.dec={:4.6f}&-c.bm={:4.7f}/{:4.7f}&-out.max=unlimited'.format(radeg,decdeg,fovam,fovam)
    #final URL: make sure it does not have any spaces or carriage returns/line feeds when copy-pasting
    sr = str1+str2

    # Read from the webpage, storing the page's contents in 's'.
    f = url.urlopen(sr)
    s = f.read()
    f.close()

    #column interpretation compass
    namecol, RAcol, DECcol, rband = 0, 1, 2, 12
    null1, null2 = '     ', ''

    #split webpage string into lines
    sl = s.splitlines()
    sl = sl[45:-1] # get rid of header
    name = np.array([])
    rad = np.array([]) # RA in degrees
    ded = np.array([]) # DEC in degrees
    rmag = np.array([]) # rmage
    #get data from each line
    for k in sl:
        kw = k.decode().split('\t')
        if kw[0] != '':
            name = np.append(name,kw[namecol])
            rad = np.append(rad,float(kw[RAcol]))
            ded = np.append(ded,float(kw[DECcol]))
            # deal with case where no mag is reported
            if (kw[rband] != null1) and (kw[rband] != null2):
                rmag = np.append(rmag,float(kw[rband]))
            else:
                rmag = np.append(rmag,np.nan)
    #return data
    return name,rad,ded,rmag

#main function
if __name__ == '__main__':

    import matplotlib.pyplot as plt
```

```
#get stars in a 5'x5' square around (RA, DEC = 30, 30)
name,rad,ded,rmag = usno(30,30,5)
print(rmag) #there's missing mags, but you don't need mags anyways.

#plot positions plt.title("USNO-
B1 Stars") plt.scatter(rad, ded,
marker='+') plt.xlabel("RA
[deg]") plt.ylabel("DEC [deg]")
plt.tight_layout()
plt.show()
```