Proper Motion of 7 Iris
2/20/2022
Mikaela Larkin
mmlarkin@ucsd.edu
Group B: Nazgol Hadaegh, Megan Li, Angelo Romvos, and Joman Wong

Abstract

Astrometry is a form of astronomical study in which a reference frame is used to track changes in proper motion of celestial objects. In this lab, we analyze the proper motion of the asteroid 7 Iris using observations from the Nickel 1-meter telescope at the Lick Observatory and the United States Naval Observatory catalog (USNO-B1.0) of known star positions as the reference frame. Automated routines were used to find the locations of bright objects in observational images and to calculate their centroid values for three of the frames taken with the Nickel 1-meter telescope: one in the R-band, one in the V-band, and another in the R-band taken one hour later. The USNO-B1.0 catalog was searched using the VizieR query to construct a region of the same size as the Nickel 1-meter telescope, 6.3x6.3 arc minutes, centered at the right ascension (RA) and declination (Dec) used to point the telescope. These values were then matched up to the centroids, converted to standard Cartesian coordinates, and run through a general least squares best-fit to account for distortions in the detector. The final comparison of the first and last R-band frames of 7 Iris displayed an increasing RA and positive, decreasing Dec, meaning the object's proper motion is in the northeast direction. This result is accurate for the general location and velocity order of magnitude, though imprecise for the exact location and direction due to remaining errors between the centroid and USNO-B1.0 locations.

Introduction

Astrometric measurements are critical to measuring the objects in motion in space, particularly the measurements for nearby asteroids to ensure that none are due to impact Earth in the near future. The main asteroid belt is located between the orbits of Mars and Jupiter, meaning the objects within the belt are not in the immediate vicinity of Earth, but the objects are close enough on an astronomical scale to observe efficiently from Earth-based telescopes. The asteroid we chose to analyze in this lab is 7 Iris, which is the fourth brightest object in the main asteroid belt. 7 Iris is a relatively large asteroid with a diameter of approximately 200 kilometers, which is why it is more visible from Earth than most of the other objects in the asteroid belt. 7 Iris is classified as a stony, or S-type, asteroid, meaning it is composed of silicates. The stony composition makes asteroids of the S-type highly reflective of sunlight, with 7 Iris at times reaching a peak brightness of about 7 magnitudes. In this lab, we tracked the proper motion of 7 Iris as it was viewed in the R-filter of the Nickel 1-meter telescope at Lick Observatory across the span of about an hour on the night of February 28th, 2022.

The process of astrometry involves using known locations of astronomical objects as a reference point to track the motion of less known objects. The reference objects tend to be stars at large distances, which move very little from our perspective on Earth. Numerous catalogs exist of known stars in the sky, including the United States Naval Observatory catalog (USNO-B1.0), which details the positions and proper motions of over 1 billion stars below 21 visual magnitudes. Later sky surveys include data for many more objects at fainter magnitudes, however, the USNO-B1.0 is sufficient for our purposes since the Nickel 1-meter telescope has a visual limit of about 18 magnitudes. Any potential objects detected that are fainter than this limit are unreliable since they may be indicative of erroneous pixels rather than actual objects. In this lab, we used a frame at the same right ascension (RA) and declination (Dec) as the images we captured with the Nickel 1-meter telescope to match known objects with stars surrounding 7 Iris. We then applied a general least squares fit to this data in order to account for any distortions and aberrations on the detector at the Nickel 1-meter telescope and to produce a final change in RA and Dec of 7 Iris across the hour of observation.

The general least squares technique accounts for four kinds of distortions that may occur in images collected by the Nickel 1-meter telescope: translation, magnification, rotation, and shear. These distortions set images collected apart from the true positions of objects in the sky as cataloged by all sky surveys. Using the mathematical technique of applying an affine transformation matrix, which consists of corrections to all four distortions, allows for resultant errors to be undone and proper sky coordinates to be found. The term general least squares is used to refer to how the matrix is applied through linear algebra techniques, which will be further described in the Calculations and Modeling section. The application of general least squares allowed for accurate, though not precise, tracking of the asteroid's location over time. The process is accurate because it can be applied to any astronomical image and corresponding catalog for a similar result. However, the calculated RA and Dec of 7 Iris are not quite precise because distortions are not the only errors present in our data. Some errors are due to physical shifts of the Nickel 1-meter telescope, as well as other detector errors that cause slight misalignment with the USNO-B1.0 catalog.

Observations and Data

In order to track the motion of an asteroid, we used the Nickel 1-meter telescope located at the Lick Observatory. Before collecting our data, we prepared for the observation by first calculating the RA range accessible by the telescope on the night of our observation, determining the range to be between approximately 2 hours and 10 hours. The RA range calculation was done assuming an hour angle of 3 hours between 12-degree twilight and 9pm, which was the local time that we projected we would be finished with our observations. The RA information, along with visible magnitude and Dec information from the Jet Propulsion Lab (JPL) small bodies website, allowed us to narrow down 7 Iris as our intended target to study. Our choice was verified as viable using the STARALT website to confirm the asteroid's visibility through the atmosphere on the observation night. We also used the Aladin website to confirm there are at least 6 stars immediately surrounding the object's RA and Dec at the time of observation to use as reference points in later calculations. The exposure time that was used for our images was calculated using the magnitude formula $m_1 - m_2 = -2.5log_{10}(\frac{F_1}{F_2})$ in which m represents visible magnitude and F represents flux. We used the star Landolt 109-954 for the reference object with subscript 2, with a visible magnitude of 12.44 mag, R-band magnitude of 11.68, and a peak flux of 1000 data numbers per second. We calculated the time for the asteroid to have a signal of 20000 data numbers in the R-band using the JPL visible magnitude of 9.06 mag, for a final result of about 1 second. In practice, however, this time needed to be increased to 1.5 seconds for the R-band and 2 seconds for the V-band since not enough light was reaching the detector in shorter time frames. Our observation plan was completed using the JPL Horizons website to find the exact RA and Dec of 7 Iris at the beginning of our observation. We pointed the telescope to this specific location since asteroids have high proper motion, thus, their coordinates change rapidly. The final image data collected can be found in the table below.

Purpose	File Name	Time collected	Personnel	Exposure time
Bias frames	d1.fits - d5.fits	February 28, 2022, ~6pm PST	Professor and TA	0 seconds
Flat frames in various filters	d6.fits - d25.fits	February 28, 2022, ~6pm PST	Professor and TA	3, 5, 25, and 120 s
R-band, first time frames	d49.fits - d54.fits	February 28, 2022, ~7:45pm PST	Group B	1 s (first frame), 1.5 s
V-band, first time frames	d55.fits - d59.fits	February 28, 2022, ~7:50pm PST	Group B	2 s
R-band, second time frames	d94.fits - d98.fits	February 28, 2022, ~9pm PST	Group B	1.5 s

In order to collect the data, we used the remote control room at UCSD with access to the Lick Observatory. The error reduction frames were collected earlier in the night, beginning with the bias frames taken with 0 exposure time and no filter to detect intrinsic error in the device. The second type were flat frames taken with each filter of the inside of the dome to account for errors present with uniform illumination of the detector. The first two sets of science frames were then collected after the sun had fully set of 7 Iris in the R-filter and in the V-filter. The third and final set of science frames was taken in only the R-band about an hour later. To collect the science frames, we pointed the telescope to the same RA and Dec of 07:10:57.8+16:07:59 for both time frames, ensured proper settings on the device for exposure time and filter, manually tracked the frames collected, and ensured that the deadman timer did not automatically close the dome at Lick Observatory. These tasks were split among the group members, with my main contribution being the person to control pointing the telescope, ensuring the settings, and collecting the images for the second round of frames in the R-band.

The reference frame data used for comparison in our astrometric measurements came from the USNO-B1.0 of known star positions in the sky. In order to access this catalog, we used the VizieR query service that searches for certain regions of the sky and builds a data structure of the desired shape around the input RA and Dec values. For our purposes, we matched this data structure to the size of visibility to the Nickel 1-meter telescope, which is 6.3x6.3 arc minutes. The RA and Dec values input were the same as those entered into the navigation of the Nickel 1-meter telescope, however, the images taken were slightly offset from this point due to small movements of the telescope. Slight telescope movement is likely a source of some offset in the images collected from catalog locations later on in our analysis, which will be described in more depth in the Discussion section.

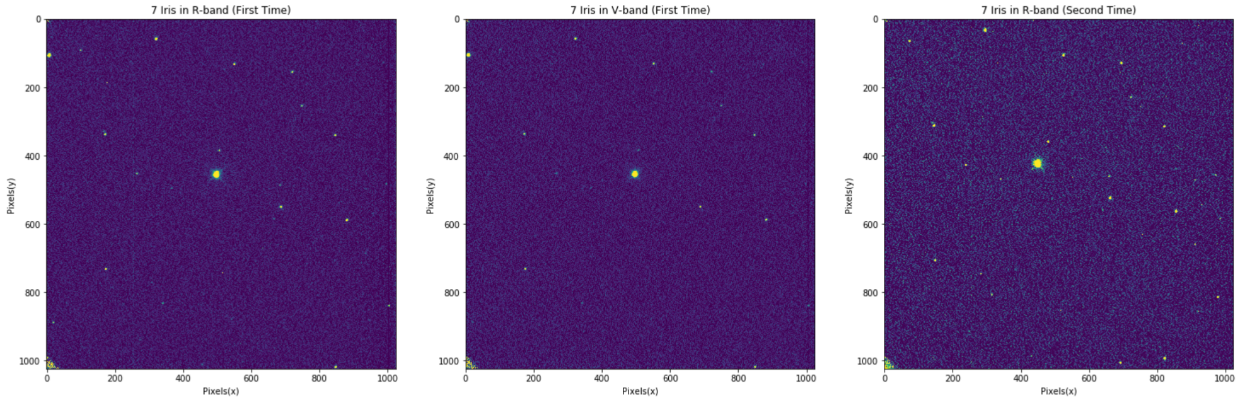
Data Reduction and Methods

In order to prepare the science data for analysis, we first had to account for potential error in the CCD similar to the previous two labs. This process began by subtracting bias data from each of the three science data frames to account for intrinsic error in the detector. Next we used the fits header information to remove the overscan columns of the CCD and to find which flat frames were taken in the V-band and R-band. The science data was then scanned for hot pixels by converting oversaturated points to have an intensity of 1 analog-to-digital unit (ADU). The data was normalized using flat frames taken with the corresponding filter by dividing by the normalization factor $F_{norm} = \frac{flat - bias}{median(flat - bias)}$. Division by zero was avoided by replacing any 0 points in the normalization factor array with a factor of 1 instead. The three science frames chosen to go through this process were one in the R-band at the first observation time, one in the V-band at the first observation time, and one in the R-band at the second observation time as shown in **Figure 1** after reduction. Celestial objects in the frame were then automatically identified by checking each pixel and the surrounding pixels for flux 2.2x brighter than the average intensity of the image through a process further described in the Calculations and Modeling section. The centroids of each object in the frame were then calculated, including the uncertainties of these centroids and the signal-to-noise ratio of each of these objects. The centroids of objects around the far edges of the detector resulted in very large uncertainties on the order of about 1000 pixels, which was likely due to distortions in the device, the rounded corners of the dome projected onto the square detector, and the partial visibility of stars on the edges of the frame. Centroids in the center of the detector had lower uncertainties on the order of about 10 pixels, which are those of 7 Iris and the immediately surrounding stars.

In the second stage of the lab, the RA and Dec input into the telescope and the shape of its field of view were used to produce a VizieR object for comparison of a reference frame of objects in the USNO-B1.0 catalog to the centroid values from the science frame. The USNO-B1.0 catalog produces results for both the RA and Dec of stars in the year 2000, as well as their proper motion in equatorial coordinates, which we converted to our time frame of 2022 and compared to the science frame coordinates. We then converted the VizieR RA and Dec to Cartesian rectangular coordinates assuming ideal conditions on the camera of only having distortions from translation and magnification to compare to the Cartesian pixel values of the science frames. A more accurate mapping was then created using a general least squares fit with matched points towards the center of the R-band frames in each time frame. The remaining difference between the frames and USNO-B1.0 was calculated to be on the order of a few pixels. The general least squares solution was applied to produce plots with all distortions minimized in order to observe the movement of 7 Iris and convert the coordinates back into RA and Dec for the asteroid. Initial and final RA and Dec were compared for a final result for 7 Iris' motion.

Calculations and Modeling

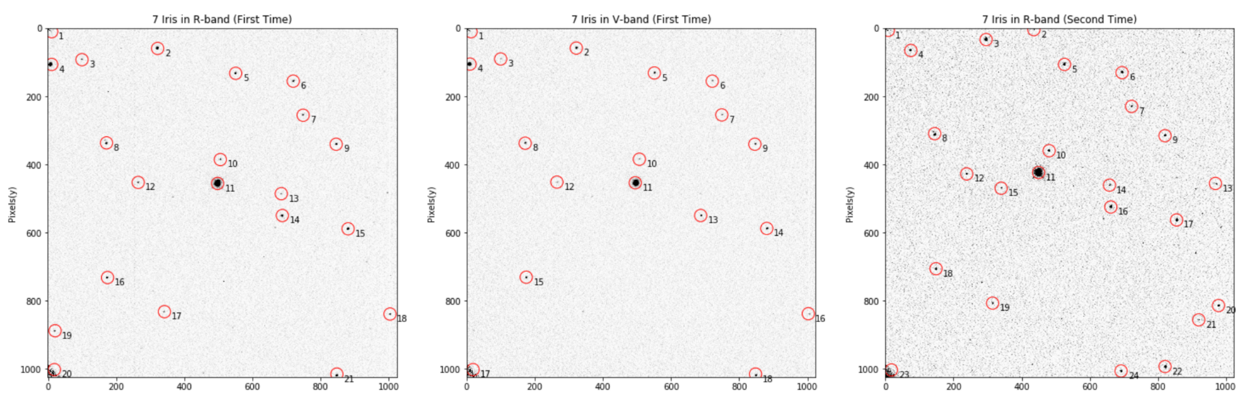
After reducing and preparing the three science frames shown in **Figure 1**, the first stage of analysis was to use an automated routine to identify celestial objects in the field of view for each frame. In order to identify objects, we created a function to go through each pixel of the



(A) **(B)** **(C)**

Figure 1: Reduced science images from the Nickel telescope of (A) the R-band in the first time frame, (B) the V-band in the first time frame, and (C) the R-band in the second time frame. The brightest object is 7 Iris.

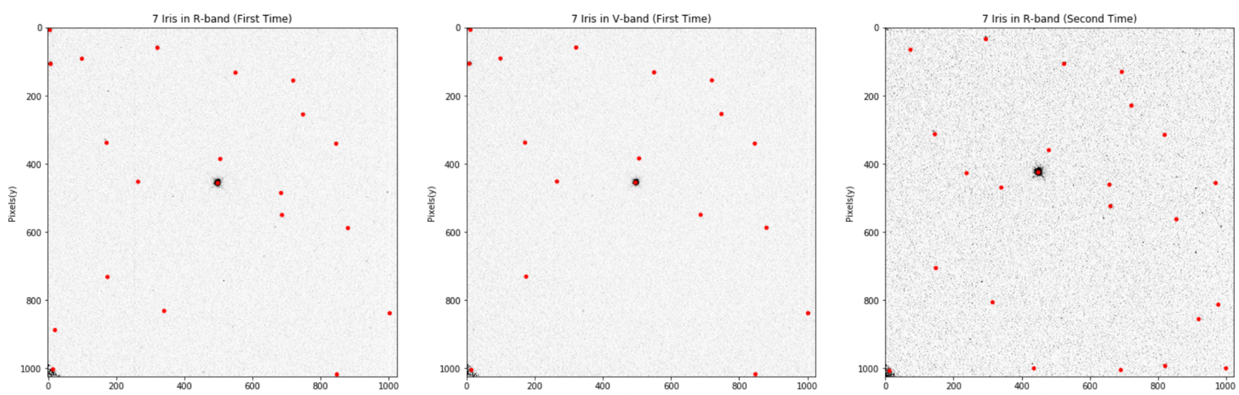
images and determine whether their intensity was above a threshold of 2.2x the mean intensity. The mean intensity value was retroactively chosen according to a later VizieR analysis described further in this section. The inputs to the function other than the science frame intensity are window size and bright pixel count threshold. If a pixel was seen as having an intensity above the minimum threshold, it was cataloged as being a star or the asteroid. The inputs to the function other than the frame are window size and bright pixel count threshold. If a pixel was determined to be bright enough to be a star or 7 Iris, other pixels within that window were checked. If more than the input pixel count threshold were also stars or the asteroid, they were appended into a list of positions. The function then continued to check through the pixels, ignoring pixels that were not bright enough to be in a star or 7 Iris, as well as those that had already been appended. The neighboring bright pixels to each object were appended into a final array of all notable object pixels. To find the approximate center of these objects for identification, the mean pixel value of each object was taken and appended into an array, which we then used to produce a final plot showing the identified objects in **Figure 2** for each of the three science images. The function used may be found in the Appendix.



(A) **(B)** **(C)**

Figure 2: Identified objects in the science frames of (A) the R-band in the first time frame, (B) the V-band in the first time frame, and (C) the R-band in the second time frame.

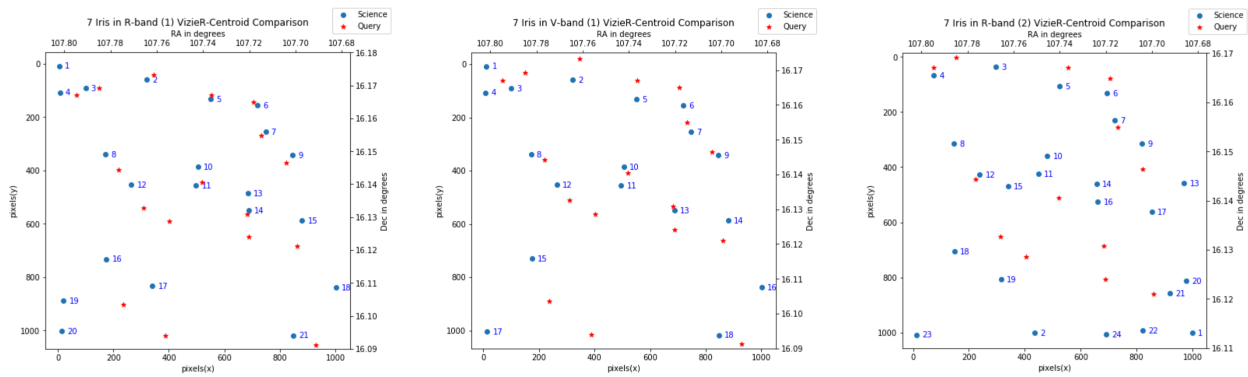
In order to find the exact location of the central intensity of each object, we implemented a function to calculate the centroids in both the x and y-direction for each object as well as their corresponding uncertainties. The centroid formula is shown by $p_c = \dfrac{\sum_{i=1}^{N} p_i I_i}{\sum_{i=1}^{N} I_i}$ in which p_c represents either the x or y pixel value for centroids, p_i represents each pixel in the peaks, and I_i represents each intensity value in the peak in analog-to-digital units (ADU). The uncertainty of these centroid values is shown by the formula $\sigma_{p_c}^2 = \dfrac{\sum_j I_j (p_j - p_c)^2}{(\sum_i I_i)^2}$ and the signal-to-noise ratio is approximated by $\sqrt{F_N}$ where F_N is the flux per counts of each object. The function to calculate centroids and their uncertainties uses the science frame intensity data, an empty position array of the same size as the science frame images, the list of locations from the previous function, and a window size for summing pixels around the object center as inputs. For each mean center of the objects previously calculated, the position array is split into windows of the specified size surrounding the central pixels, appending all of the pixel values in the window into arrays of x and y-values. For the x and y-directions separately, the centroid formula specified previously is calculated and any centroid with a possible division by zero is replaced by an arbitrary value of 1000. Similarly, the uncertainty values for each direction are calculated with the same avoidance of division by zero. The final result is an array of centroid values and uncertainties in the x and y-directions, with the final centroids shown in **Figure 3** as points on the three science frames. The uncertainty values and signal-to-noise ratios will be analyzed further in the Discussion section.



(A) **(B)** **(C)**

Figure 3: Centroid positions in the science frames of (A) the R-band in the first time frame, (B) the V-band in the first time frame, and (C) the R-band in the second time frame.

The next stage of analysis consisted of using the USNO-B1.0 catalog at the same RA and Dec as the science frames to compare them with a reference frame using the VizieR catalog query. The specific RA and Dec input was 07:10:57.8+16:07:59, which are the coordinates we used to point the telescope in the observation phase of the lab. The Nickel 1-meter telescope has a visual magnitude limit of approximately 18mag, but this input identified too many objects in the 6.3x6.3 arc minute field of view visible to the telescope, so we set a magnitude limit of 15.5 instead. The limit of 15.5 identified 21 objects in the USNO-B1.0 catalog, which we went back and used to inform our choice of 2.2x the mean intensity in our previous function to locate objects. The epoch difference between the year 2000, when USNO-B1.0 was set, and the current



(A) **(B)** **(C)**

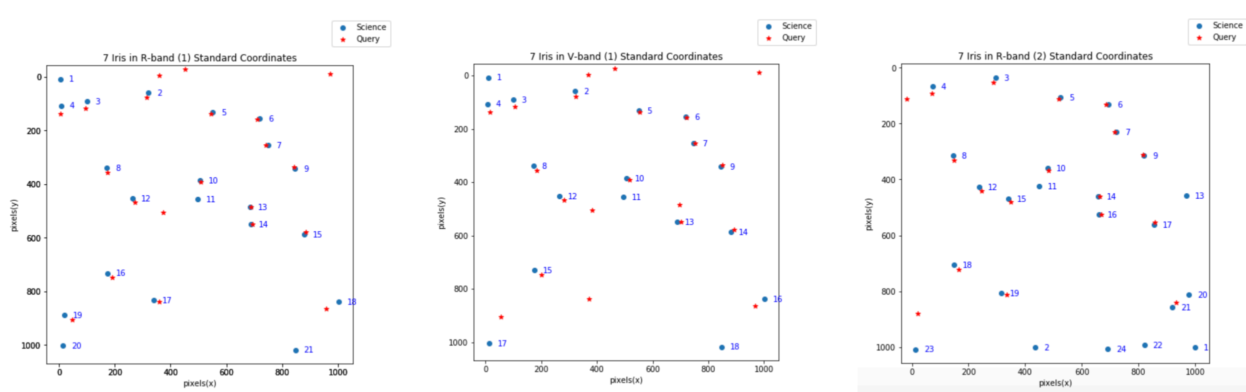
Figure 4: Direct comparison of VizieR RA and Dec to science frame pixel positions for (A) the R-band in the first time frame, (B) the V-band in the first time frame, and (C) the R-band in the second time frame.

year 2022, when our data was taken, had to then be reconciled. We accounted for the difference by converting the VizieR output for the proper motion RA and Dec to units of degrees per year and multiplying the results by 2022 - 2000 = 22. These values were added to each object's RA and Dec result in the region of the frame. The final VizieR RA and Dec were then directly compared to the centroid locations calculated for our three science frames as shown in **Figure 4**.

To properly match up the VizieR points with the centroid locations, the VizieR RA and Dec first needed to be converted to standard Cartesian coordinates according to the formulas

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

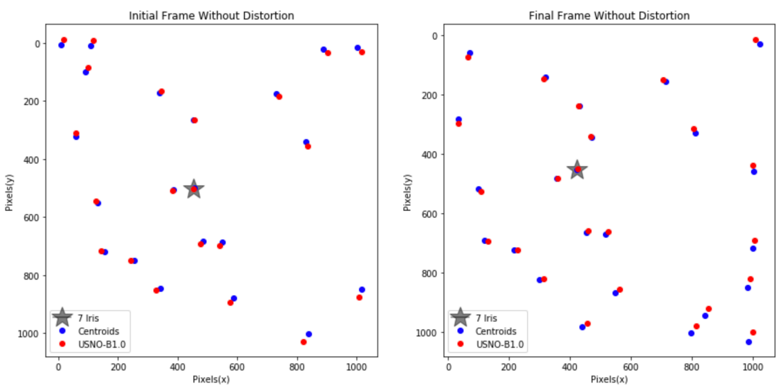
where X and Y are rectangular coordinates, α_0 and δ_0 are expected RA and Dec from the observational input in radians, and α and δ are RA and Dec from VizieR in radians. These coordinates were then compared to the science frame CCD pixel locations according to the formulas $x = f * \frac{X}{p} + x_0$ and $y = f * \frac{Y}{p} + y_0$ for an ideal camera for pixel coordinates x and y, focal length of the Nickel 1-meter telescope f = 16840 mm, and pixel size on the CCD p = 0.015 mm.



(A) **(B)** **(C)**

Figure 5: Standard Cartesian coordinate comparison of ideal VizieR points to science frames for (A) the R-band in the first time frame, (B) the V-band in the first time frame, and (C) the R-band in the second time frame.

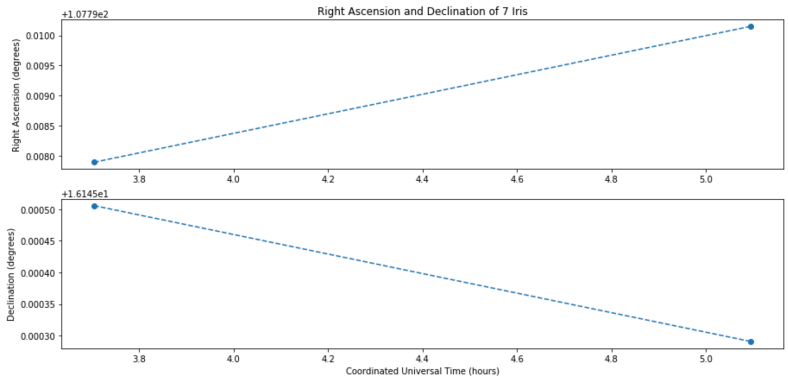
These formulas assume the camera at Nickel to be ideal, meaning the only effects of distortion are due to translation and magnification. The plots of each frame and corresponding USNO-B1.0 objects under the ideal assumption are shown in **Figure 5**. The ideal solutions for x and y are temporarily used in order to define the more complex distortions on the detector derived in the next step.



(A) **(B)**

Figure 6: General least squares comparison in the R-band of VizieR objects and centroids without distortions to track proper motion of 7 Iris in (A) the first time frame and (B) the second time frame.

The centroid values and corrected USNO-B1.0 values were then matched up to each other by calculating the minimum distance between objects from each catalog and appending those within an arbitrary value of 30 pixels of each other for the first frame and 27 pixels for the third. These values were chosen to have the same number of 10 matched up points across the two time frames, including the point of 7 Iris. The second frame in the V-band was not used from this point forward because the tracking of the motion of 7 Iris over time was conducted in the same filter, the R-band. In order to find the plate constants a_{11}, a_{11}, a_{11}, and a_{11} for the CCD at Nickel as well as the translational pixel constants x_0 and y_0 from the previous mapping of the USNO-B1.0 objects with coordinates X and Y to their corresponding centroid points in the science frame, the formulas $\begin{pmatrix} x_1 \\ x_2 \\ \cdots \\ x_N \end{pmatrix} = \begin{pmatrix} \frac{f}{p}X_1 & \frac{f}{p}Y_1 & 1 \\ \frac{f}{p}X_2 & \frac{f}{p}Y_2 & 1 \\ \cdots & \cdots & \cdots \\ \frac{f}{p}X_N & \frac{f}{p}Y_N & 1 \end{pmatrix} (a_{11} \quad a_{12} \quad x_0)$ and $\begin{pmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{pmatrix} = \begin{pmatrix} \frac{f}{p}X_1 & \frac{f}{p}Y_1 & 1 \\ \frac{f}{p}X_2 & \frac{f}{p}Y_2 & 1 \\ \cdots & \cdots & \cdots \\ \frac{f}{p}X_N & \frac{f}{p}Y_N & 1 \end{pmatrix} (a_{21} \quad a_{22} \quad y_0)$ were used with the same variables as previous formulas, where indexed x and y are the values from the ideal solution. The plate constants define all distortions on the camera including beyond the ideal camera approximation by accounting for rotation and shear in addition to translation and magnification. Once the plate constants were calculated for the device using the first R-band frame, the final affine transformation matrix $\bar{T} = \begin{pmatrix} \frac{f}{p}a_{11} & \frac{f}{p}a_{12} & x_0 \\ \frac{f}{p}a_{21} & \frac{f}{p}a_{22} & y_0 \\ 0 & 0 & 1 \end{pmatrix} \approx \begin{pmatrix} 0.998 & -0.018 & 15.178 \\ -0.002 & 0.946 & 32.531 \\ 0 & 0 & 1 \end{pmatrix}$ was used to create a full map of all centroid points, including 7 Iris, which accounted for all distortion as shown in **Figure 6**. This matrix was also applicable to the second time frame since it describes the physical nature of the device itself. The way the matrix was applied to centroid values was according to the general least squares formula $\bar{X} = (\bar{T}^T\bar{T})^{-1} * \bar{T}^T * \bar{x}$ in which $\bar{X}$ are the desired standard coordinates, $\bar{T}$ is the transformation matrix, and $\bar{x}$ are the centroid values. 7 Iris was identified by using the index near the center of the frame that initially had no equivalent in the USNO-B1.0 catalog with the most apparent motion across the two frames. The remaining difference between the matched up points of 7 Iris were calculated to be about x = 3.2 and y = 3.9 pixels in the first frame and about x = 1.9 and y = 2.1 pixels in the second frame.



(A)

(B)

Figure 7: Final comparison of the proper motion of 7 Iris in terms of (A) right ascension and (B) declination across the hour measured on February 28, 2022.

To produce the final proper motion tracking of 7 Iris, the standard coordinates X and Y calculated in the previous step were converted to RA and Dec using the formulas

$tan(\alpha - \alpha_0) = -\dfrac{X}{cos(\delta_0) - Ysin(\delta_0)}$ and $sin(\delta) = \dfrac{sin(\delta_0) + Ycos(\delta_0)}{\sqrt{1 + X^2 + Y^2}}$ using the observational center of the frame for reference RA and Dec. The change of RA and Dec in degrees defines the change in proper motion of 7 Iris as shown in **Figure 7**. Right ascension is shown to increase by about $1.625 * 10^{-3}$ degrees per hour and declination is shown to decrease by about $1.547 * 10^{-4}$ degrees per hour, meaning the asteroid is moving in the northeast direction. This trend occurs because right ascension increases eastward, and since the asteroid has a positive declination value, the star moves north as Dec decreases. This motion is shown as it appears in the sky to the Nickel 1-meter telescope in **Figure 8**.

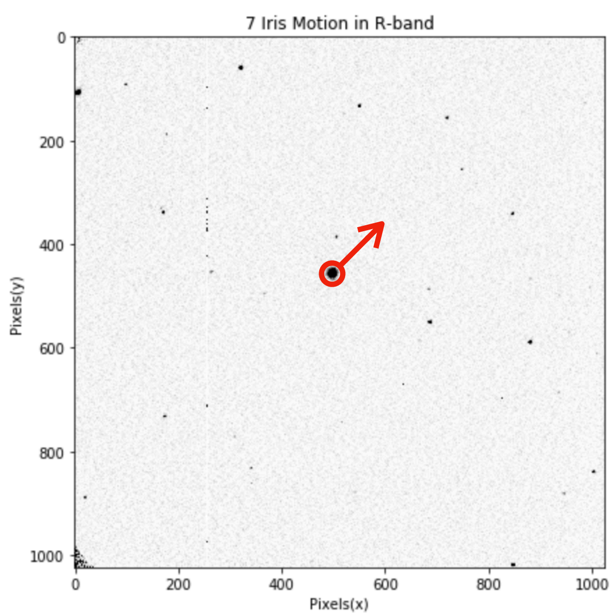


Figure 8: Final result based on our calculations for the direction of motion of 7 Iris across the plane of the sky in the view of the Nickel 1-meter telescope.

Discussion

The overall results for the RA and Dec of 7 Iris indicate that there is no imminent concern over the motion of the asteroid to life on Earth since it is located completely within the orbit of the main asteroid belt at a relatively low orbital velocity. The calculated results for the proper motion of 7 Iris indicate motion in the northeast direction with an increasing RA and decreasing, positive Dec. The rate of change in each direction, $1.625 * 10^{-3}$ RA degrees per hour and -$1.547 * 10^{-4}$ Dec degrees per hour, is relatively small considering the overall scale of 360 total degrees around the sky. The RA and Dec results translate to a change of about 00:00:00.3 RA in hours, minutes, and seconds, and about -00:00:00.6 Dec in degrees, arc minutes, and arc seconds, over the measured time span. The JPL Horizons predictions for change in RA and Dec over the Lick Observatory in this time frame were about 00:00:00.5 RA and +00:00:01.5 Dec, so our results were accurate for RA, but were inaccurate for declination. JPL Horizons projects declination to be increasing over this period, meaning the object was moving south rather than north in the sky. The reason for this incorrect result is likely due to the misalignment between pixels of the detector and the final Cartesian results of USNO-B1.0 catalog of about 3.5 pixels in the fits frame and of about 2 pixels in the second frame. This error was likely caused by the shifting of exact RA and Dec when pointing the telescope. The precise point at which the telescope was pointing was not exactly the same as the navigation input and the result was different between the first and second frames. Since the motion of the asteroid is so small in relation to the plane of the sky, any small offset between captured frames can skew the astrometric calculations. In future observations, telescopes used for observations should be completely stabilized for accurate measurements of the low-velocity celestial objects.

The RA and Dec results prove to be imprecise in regard to the exact location and direction of 7 Iris due to the shifting of the telescope. Another cause for imprecision is device errors, such as the rounded corners on the field of view of the Nickel 1-meter telescope. These device errors were detected by the star locating algorithm due to their high intensity and had to be excluded in the general least squares step, along with heavily distorted stars near the edges. Once the index of 7 Iris was determined, the signal-to-noise ratio calculated for the first frame in the centroid-finding routine could be assessed for the asteroid. The final result was approximately 95.4 in the x-direction and 446.1 in the y-direction, for an average result of about 270.8. This is a very high signal-to-noise ratio, meaning the images taken with the Nickel 1-meter telescope were very clear for the object, making the result more accurate in regard to the systematics of the detector. The final RA and Dec for 7 Iris also indicates accuracy due to the order of magnitude for the speed at which the asteroid is moving compared to the speed projected by JPL Horizons. In conclusion, the exact sky coordinates of 7 Iris were calculated with low precision, but high accuracy. The study of this asteroid could be improved by using a more stable telescope with a higher magnitude limit and a reference frame from a more advanced sky survey such as Gaia, which includes 0.8 billion more sources than USNO-B1.0 and some information about the parallax distance to different objects. However, in future studies, the process of general least squares fitting to track celestial objects will still provide accurate results since it accounts for all four major sources of distortion. If 7 Iris eventually begins to appear brighter than the current 7 visual magnitudes with more rapid change in coordinates some day, further analysis of its trajectory using more precise astrometric measurements would be necessary to predict its future.

Appendix

All five of us in the lab group worked collaboratively in class to understand the lab and to assemble our observation plan. In the observation run, we took turns testing the various operations necessary for the Nickel 1-meter telescope. After this point, our lab group operated using group messaging to answer each other's questions about the lab as they arose. Joman, Angelo, Megan, and I discussed the data reduction portion of the lab to figure out how to access the fits files, remove the overscan region, and perform the bias subtraction. The automated routine to locate bright objects was discussed by Megan and me until I was able to get the routine working and contributed my function, which is shown below. Similarly, I also contributed my centroid finding routine shown below after discussion with Megan. Joman and I answered questions about how to set up the initial comparison to the Vizier objects. I contributed my code to convert to ideal Cartesian coordinates shown below. I also explained the use of images in the V-band as definitive of the magnitude limit to the group. Megan helped me to set up the final conversion to RA and Dec for 7 Iris.

The sources for observation planning were the JPL small bodies website https://ssd.jpl.nasa.gov/sbwobs.cgi, STARALT http://catserver.ing.iac.es/staralt/index.php, Aladin http://aladin.u-strasbg.fr/, and JPL horizons https://ssd.jpl.nasa.gov/horizons/app.html#/.

Information was found for 7 Iris at https://en.wikipedia.org/wiki/7_Iris and S-type asteroids at https://en.wikipedia.org/wiki/S-type_asteroid.

Conversions from output degrees to hours, minutes and seconds for RA was done with https://www.vercalendario.info/en/how/convert-ra-degrees-hours.html and to degrees, arc minutes, and arc seconds for Dec with https://rechneronline.de/winkel/degrees-minutes-seconds.php.

Function for removing overscan region and subtracting bias:

```
def load_frame_overscan_remove_bias_subtract(filename):
    """
    ARGUMENTS:
    ===============================================================
    filename - Name of any frame intended for reduction

    RETURNS:
    ===============================================================
    Image data points with bias subtracted and overscan region removed
    """

    hdr = fits.open(filename)           # Opening desired file
    overscan = hdr[0].header['COVER']   # Obtaining length of overscan region

    image = fits.getdata(filename)      # Extracting data points from file
    image = image - fits.getdata(bias)  # Subtracting bias frame from science
    image = image[:,:-overscan]         # Removing overscan region from the end of data points

    return image
```

Function for data reduction for science frames:

```python
def load_reduced_science_frame(filename, flat):
    """
    Loads a Science frame specified by the FITS file name,
    performs reduction through bias subtraction and normalized
    flat field correction and returns the data

    ARGUMENTS:
    ================================================================
    filename - Name of science frame intended for reduction
    flat - Name of flat file using corresponding filter

    RETURNS:
    ================================================================
    Reduced science frame with bias subtracted, overscan removed, and flat normalization applied
    """

    sdata = load_frame_overscan_remove_bias_subtract(filename)
                    # Running science frame through previous function to bias subtract and remove overscan
    sdata[sdata >= 60000] = 1          # Removing hot pixels from science frame data
    fdata = load_frame_overscan_remove_bias_subtract(flat)
                    # Running flat frame through previous function to bias subtract and remove overscan
    fnorm = fdata / np.median(fdata)  # Normalization factor
    fnorm[fnorm == 0] = 1             # Removing values of 0 from normalization factor
    data = sdata / fnorm              # Final form of data after reduction

    return data
```

Function for identifying bright objects in science frames:

```python
def find_star_locs(im_data, n_size = 10, bright_count_thresh=10):
    """
    ARGUMENTS:
    ================================================================
    im_data - Reduced 2D FITS Data
    n_size  - Neighborhood size to consider for windowing (pixels)
    bright_count_thresh - Threshold for number of 'bright'pixels in
                          neighborhood to be considered a star.
    RETURNS:
    ================================================================
    x and y positions of stars in science frame
    """
    mew = np.mean(im_data)
    thresh = mew * 2.2           # Minimum threshold for intensitiy to be considered a star

    neighbors = [                # Region to check surrounding pixels for star intensity
        [-1, -1],
        [-1, 0],
        [-1, 1],
        [0, -1],
        [0, 1],
        [1, -1],
        [1, 0],
        [1, 1]
    ]

    num_rows, num_columns = im_data.shape         # Separating size of input data

    star_id = 1
    stars = np.zeros((num_rows, num_columns, 2))  # Empty array with 2 channels: threshold met and star identification

    star_pixels_dict = {}

    for i in range(num_rows - n_size):
        for j in range(num_columns - n_size):
            window = im_data[i:i+n_size, j:j+n_size]          # Input size of area to check for stars
            window_thresh = np.where(window >= thresh, 1, 0)
            num_brights = np.sum(window_thresh)

            if num_brights > bright_count_thresh:  # Regions with more star pixels than bright_count_thresh input
                x = i + n_size//2
                y = j + n_size//2
                stars[x, y, 0] = 1
```

```
    for i in range(num_rows):
        for j in range(num_columns):
            if stars[i, j, 0] == 0 or stars[i, j, 1] != 0: # If pixel is not in a star or is in previously seen star
                continue

            # Is a new star
            star_pixels = [[i, j]]
            stars[i, j, 1] = star_id
            star_pixels_dict[star_id] = []                # Dictionary to refer to indices of stars

            while len(star_pixels) > 0:                   # While there are pixels in the star
                x, y = star_pixels.pop(0)
                star_pixels_dict[star_id].append([x, y])
                for n_x, n_y in neighbors:
                    n_x += x                              # Adding the neighboring pixels in stars
                    n_y += y

                    if stars[n_x, n_y, 0] == 1 and stars[n_x, n_y, 1] == 0:
                        # Add star to list of all stars
                        star_pixels.append([n_x, n_y])
                        stars[n_x, n_y, 1] = star_id

            star_id += 1  # Identifying stars by numbers starting with 1

    list_pos = []
    for star_id, star_pixels in star_pixels_dict.items():
        min_x = im_data.shape[0] + 100000          # Starting with a large number to later find minimum pixels
        max_x = -1
        min_y = im_data.shape[1] + 100000
        max_y = -1

        for x, y in star_pixels:
            min_x = min(min_x, x)
            max_x = max(max_x, x)
            min_y = min(min_y, y)
            max_y = max(max_y, y)

        center_x = (min_x + max_x) // 2
        center_y = (min_y + max_y) // 2
        list_pos.append([center_x, center_y])       # Appending integer values for center of stars

    star_centers = np.zeros(im_data.shape)          # Empty array to fill with center values
    for x, y in list_pos:
        padd = n_size // 2
        star_centers[x-padd:x+padd, y-padd:y+padd] = 1  # Appending star centers within input size

    return list_pos
```

Function for calculating centroids, uncertainties, and signal-to-noise ratio:

```
def calc_centroids_2d(intarr, posarr, loc_list, window_max = 20):
    """
    Calculating centroid location for each star in frame according to equation above

    PARAMETERS:
    ==================================================================================
    intarr - array of intensity for each star
    posarr - 2d array with same dimensions as frame
    loc_list - array of star locations in frame
    window_max - Size of Window to consider to find max pos of each star (in pixels)

    RETURNS:
    ==================================================================================
    centroids - List of list of centroid coordinates and corresponding uncertainities
                Format: [[xc, yc, unc_xc, unc_yc]]
    """

    centroids = [] # Append Tuples (xc, yc, xc_err, yc_err) of centroid coordinates
    # No. of tuples should equal length of loc_list

    xc, yc, unc_xc, unc_yc = [], [], [], []

    for x, y in loc_list:                           # Windows around star center locations from previous function
        padd = window_max // 2
        posarr_x_window = pos_arr[x - padd:x + padd, y]
        posarr_y_window = pos_arr[x, y - padd:y + padd]
        intarr_x_window = intarr[x - padd:x + padd, y]
        intarr_y_window = intarr[x, y - padd:y + padd]
```

```
        posarr_x = [p[0] for p in posarr_x_window]  # Pixels positions in star windows
        posarr_y = [p[1] for p in posarr_y_window]

        numerator_x = 0
        denominator_x = 0
        snr_x = 0
        for i in range(len(posarr_x)):              # Centroids in x-direction
            numerator_x += posarr_x[i] * intarr_x_window[i]
            denominator_x += intarr_x_window[i]
            snr_x += np.sqrt(posarr_x[i] / window_max)
        print('SNR in x =', snr_x)              # Signal-to-noise ratio in x of asteroid

        if denominator_x != 0:
            xc = numerator_x/denominator_x
        else:
            xc = 1000  # Avoiding divide by zero by setting large number to replace those centroid values

        err_numerator_x = 0
        err_denominator_x = 0
        for i in range(len(posarr_x)):              # Centroid uncertainties in x-direction
            err_numerator_x += intarr_x_window[i] * (xc**2)
            err_denominator_x += (intarr_x_window[i])**2

        if denominator_x != 0:
            unc_xc = err_numerator_x/err_denominator_x
        else:
            unc_xc = 1000  # Avoiding divide by zero

        numerator_y = 0
        denominator_y = 0
        snr_y = 0
        for i in range(len(posarr_y)):              # Centroids in y-direction
            numerator_y += posarr_y[i] * intarr_y_window[i]
            denominator_y += intarr_y_window[i]
            snr_y += np.sqrt(posarr_y[i])
        print('SNR in y =', snr_y)              # Signal-to-noise ratio in x of asteroid

        if denominator_y != 0:
            yc = numerator_y/denominator_y
        else:
            yc = 1000     # Avoiding divide by zero

        err_numerator_y = 0
        err_denominator_y = 0
        for i in range(len(posarr_y)):              # Centroid uncertainties in y-direction
            err_numerator_y += intarr_y_window[i] * (yc**2)
            err_denominator_y += (intarr_y_window[i])**2

        if denominator_y != 0:
            unc_yc = err_numerator_y/err_denominator_y
        else:
            unc_yc = 1000  # Avoiding divide by zero

        centroids.append([xc, yc, unc_xc, unc_yc])  # Appending centroid values and uncertainties in x and y-direction

    return centroids
```

Code for converting VizieR RA and Dec to Cartesian coordinates:

```
# RA and Dec to Cartesian for Vizier objects using SkyCoord output as reference according to formula below
d = cos(skydec) * cos(dec_cat) * cos(ra_cat - skyra) + sin(skydec) * sin(dec_cat)
X = - cos(dec_cat) * sin(ra_cat - skyra) / d
Y = - (sin(skydec) * cos(dec_cat) * cos(ra_cat - skyra) - cos(skydec) * sin(dec_cat)) / d
```

Code for ideal coordinate conversion:

```
f = 16480       # Focal length in mm
p = 0.015 * 2   # CCD pixel size in mm, doubled to account for binning

# Cartesian coordinate output for VizieR objects according to formula below
# (Assumes ideal camera and same size for each of the three science frames)
x = f * (X / p) + (np.shape(science_data_1)[0] / 2)
y = f * (Y / p) + (np.shape(science_data_1)[1] / 2)
```

Code for matching up VizieR standard coordinates with centroids:

```
for i, val in enumerate(x1c):  # Matching up centroid values with corresponding ideal USNO objects for first frame
    dist = np.sqrt((x1c[i] - x1id)**2 + (y1c[i] - y1id)**2)
    mind = np.min(dist)   # Minimizing distance between points
    if mind > 30:         # Within threshold of 30 pixels to match up 10 best points
        continue
    x1_cent.append(x1c[i])  # Final centroid values
    y1_cent.append(y1c[i])
    X1_usno.append(x1id[dist == mind])  # Final USNO values
    Y1_usno.append(y1id[dist == mind])
```

Code for creating affine transformation matrix:

```
# Calculating plate constants for the device using VizieR and first frame coordinates according to equations below
# in order to later build the matrix above for the general least squares fit of the asteroid location

a_x1 = np.transpose(np.matrix(x1_cent))  # Matrix of x-values of centroids
B_x1 = np.transpose(np.matrix([(X1_usno).T[0], (Y1_usno).T[0], (np.full(len(x1_cent), 1))]))  # USNO matrix
                                # (f/p factor already absorbed into X1_usno and Y1_usno)
B_x1t = np.transpose(B_x1)      # Transpose of USNO matrix
BTB_x1 = B_x1t * B_x1           # Transpose times original USNO matrix
x1_inv = np.linalg.inv(BTB_x1)  # Inverse of BTB_x1
x1 = (x1_inv * B_x1t * a_x1).T  # Final plate constants [a_11 a_12 x_0]
x1 = np.array(x1)

a_y1 = np.transpose(np.matrix(y1_cent))  # Matrix of x-values of centroids
B_y1 = np.transpose(np.matrix([(X1_usno).T[0], (Y1_usno).T[0], (np.full(len(y1_cent), 1))]))  # USNO matrix
B_y1t = np.transpose(B_y1)      # Transpose of USNO matrix
BTB_y1 = B_y1t * B_y1           # Transpose times original USNO matrix
y1_inv = np.linalg.inv(BTB_y1)  # Inverse of BTB_y1
y1 = (y1_inv * B_y1t * a_y1).T  # Final plate constants [a_21 a_22 y_0]
y1 = np.array(y1)

T1 = np.matrix([x1[0], y1[0], np.array([0,0,1])])  # Final transformation matrix
```

Code for calculating final RA and Dec of 7 Iris in each frame:

```
# Rearranged equations above to calculate first and final RA and Dec of 7 Iris

RA1 = arctan(-X1ast / (cos(skydec)-(Y1ast*sin(skydec)))) + ((skyra*np.pi)/180)
Dec1 = arcsin((sin(skydec)+(Y1ast*cos(skydec))) / np.sqrt(1+(X1ast**2)+(Y1ast**2)))

RA3 = arctan(-X3ast / (cos(skydec)-(Y3ast*sin(skydec)))) + ((skyra*np.pi)/180)
Dec3 = arcsin((sin(skydec)+(Y3ast*cos(skydec))) / np.sqrt(1+(X3ast**2)+(Y3ast**2)))

t1 = 3.7053  # Exact time  of frames from log sheet in UTC hours
t3 = 5.0947
```