UT Austin – Department of Astronomy

Lab: CCD Astrometry – Proper Motion

1 Overview

Astrometry observations are concerned almost entirely with the positions of celestial objects in the sky. Most sub-disciplines within astronomy involve gleaning information from the light we receive to determine temperature, composition, relative motion, etc. Astrometry, however, involves just the simple position of the object. Astrometry includes measuring the positions of objects in the solar system, such as planets or asteroids, and determining an orbit. However, positions of extrasolar objects can also be determined. In this lab, we will be measuring the motion high proper motion stars.

2 Introduction

Proper motion refers to angular motion of a star in the sky due to its motion through space. All stars are in motion through space with some unknown space velocity. Stars with high proper motion are generally either nearby (so their space velocity manifests itself in their angular velocity) or are special high velocity stars such as stars in the halo.

Proper motion is measured in arcseconds per year. It is often separated into displacement in both RA and DEC though it is not listed as such in the LHS (Luyten Half-Second) catalog. It is best to think of proper motion as the motion of the star against the "stellar background" or against the coordinate grid itself.

3 Data at the Telescope.

The Luyten Half Second catalog constitutes a list of stars with high proper motion. You will need to choose a star that is visible for the current time of year. Ideally, this will be a star of which a first epoch image is available. A first epoch image is one taken at an earlier time, so that the motion will be detectable compared to the later image. After choosing your star, and finding it in the CCD camera, take several images at different exposure times, all without filters.

If the star is a binary, you will want to use a low exposure time so that the two stars do not bleed into one another, which will make it difficult for the computer to determine their centroids.

After having taken images of your high proper motion star, FTP them to a lab computer on the thirteenth floor (for which you should have accounts on if you're enrolled in the lab), or save them to a ZIP disk.

Obviously, it is impossible to determine the motion of a star with merely one night's observations. Ideally, you will be able to use first epoch images taken at some known date in the past. If not, you will have to return to the telescope late in the semester and image your star again. If it has sufficient proper motion, its motion may be detectable over the course of a semester.

4 IRAF Analysis

4.1 Method and Overview.

It will be necessary to use IRAF on the lab computers in RLM for this lab. To do this, you will need to boot the computers into Linux. It is assumed you will have some experience in IRAF itself, but not necessarily in the packages that you will need to do this lab.

With two images of the same (at least to the eye) star field taken at different times, at least a few months apart, you will be able to calculate the proper motion of the target star over time. The method by which you will be determining proper motions is to use "anonymous" reference stars in the field and track the motion of the high proper motion against these stars of assumed low proper motion. We assume that these reference stars are far enough away that their motion is not detectable. This assumption can and should be checked. That process will be discussed in a later section.

It is desirable to use guide stars – stars for which positions are accurately known by catalog – such as HIPPARCOS stars, stars from the Hubble Guide Star Catalog, or the SAO catalog. However, attempting to figure out which stars in the field match stars in various catalogs is not an easy feat. You are welcome to try and use guide stars and known RA and DEC positions; the methods listed here should be adaptable.

The concept here is to match the anonymous stars in the field (any star BUT your target star) and calculate a linear transformation between the two sets of arbitrary coordinates. That is, if you have one star at position (x_1, y_1) on image one, and that same star at (x_2, y_2) on image two, you want to create a mathematical transformation from (x_1, y_1) to (x_2, y_2).

The transformation will consist of a shift, a rotation, and a small scaling factor. IRAF will do most of the work for you. Once you have a transformation based on the reference stars, you can transform the coordinates of your target high proper motion star and see how many pixels it has moved. This, combined with the plate scale (which you will also calculate later in this lab) will produce the proper motion, which you can compare to value listed in the LHS catalog.

4.2 Using IRAF – the basics

IRAF is a collection of packages used to analyze and reduce astronomical data. While there are some graphical elements, the vast bulk of the IRAF interface is command-line. When entering IRAF, you will be greeted with the following:

```
Welcome to IRAF.  To list the available  commands, type ? or ??.  To get
detailed information about a command, type `help command'.   To  run  a
command  or  load  a  package,  type  its name.   Type  `bye' to exit a
package, or `logout' to get out of the CL.   Type `news'  to  find  out
what is new in the version of the system you are using.   The following
commands  or packages  are currently  defined:



+-------------------------------------------------------------+
|       Space Telescope Science Data Analysis System          |
|          STSDAS Version 2.1.1, December 07, 1999            |
|                                                             |
|   Space Telescope Science Institute, Baltimore, Maryland    |
|         For help, send e-mail to help@stsci.edu             |
+-------------------------------------------------------------+
ared.       dbms.       images.     nmisc.      proto.      tables.
color.      finder.     language.   noao.       softools.   utilities.
ctio.       ice.        lists.      obsolete.   stsdas.     xray.
dataio.     icex.       mcdlocal.   plot.       system.

cl>
```

The cl> prompt is where you enter your command. The list above the prompt is of various IRAF packages. Typing one of their names will then bring up another list of tasks and sub-packages. Before you execute a task, you will usually want to edit the task parameters. You do this by typing '`epar` *`task-name`*'. The edit parameters dialog is the same for all IRAF packages, but it consists of a command-line interface also. These will be explained in more detail as you need to use them.

Images are displayed in an X windows program known as *ximtool*. It is not really part of IRAF, but IRAF will use it to display images. It must be launched before you enter IRAF. Also, *ximtool* requires X windows to be running in 8 bit color depth (meaning each pixel on the screen has eight bits of information or 2^8 possibilities for color [256 colors]). If *ximtool* fails to run because of this problem, contact the system administrator. At times, when you will need to display your image, you will use the IRAF task *display*. With *ximtool* running type '`display` *`imagename`*' at the `cl>` prompt. Accept all the default options and your image will be displayed.

4.3 Arcsecond star hopping.

Through inspection of your CCD images, you will need to pick a handful of stars in both fields that are in fact the same stars. Be sure to distribute them well across the CCD image – don't pick 5 stars all in one corner. Finding matching stars may be somewhat difficult. Keep in mind that the images are most likely at different orientations. You will need to remember which stars you chose, so making a printout and circling or marking your stars may be a good idea.

To get the pixel coordinates for these stars you will need to use IRAF and the task *center*, which is under the IRAF package `noao/digiphot/apphot`. To access this package, simply type *noao*, then Enter; *digiphot*, then Enter; *apphot*, then Enter.

The center package will calculate the coordinates for a star in pixels of the image. It does this by taking starting point that you select and comparing the intensity value of the surrounding pixels, moving towards the center of the highest intensity pixels and repeating until the center pixel is the one of highest intensity, thus finding the center of the star. The coordinates for that pixel are then assigned to the star as its coordinates in the pixel frame.

With *ximtool* displaying the CCD image of the star field, type *`epar center`* at the `cl>` prompt to edit the parameters of the *center* task. The following standard text based interface will appear in IRAF when you attempt to edit the parameters of any IRAF package or task. The following is the edit parameters display for the *center* task.

```
I R A F
                    Image Reduction and Analysis Facility
PACKAGE = apphot
   TASK = center

image   = sao70919_61CygA_LHS62.fit  Input image
(coords =                      ) Coordinate list
(output =               default) Results file
(plotfil=                      ) Output plot metacode file
(datapar=                      ) Data dependent parameters
(centerp=                      ) Centering parameters
(interac=                   yes) Mode of use
(radplot=                    no) Plot radial profile plots in interactive mode
(verify =                   yes) Verify critical parameters in non-interactive
mo
(update =                    no) Update critical parameters in non-interactive
mo
(verbose=                    no) Print messages in non-interactive mode
(graphic=              stdgraph) Graphics device
(display=              stdimage) Display device
(icomman=                      ) Image cursor: [x y wcs] key [cmd]
(gcomman=                      ) Graphics cursor: [x y wcs] key [cmd]
(mode   =                    ql)
```

`image =` refers to the name of your image file.

Most of the default settings should be sufficient. You may need to make changes under centering parameters. To do this, arrow down to `centerp=` and type `:e` and you will see the following.

```
PACKAGE = apphot
   TASK = centerpars

(calgori=             centroid) Centering algorithm
(cbox   =                   5.) Centering box width in scale units
(cthresh=                   0.) Centering threshold in sigma above background
(minsnra=                   1.) Minimum signal-to-noise ratio for centering
algo
(cmaxite=                   10) Maximum number of iterations for centering
algor
(maxshif=                   2.) Maximum center shift in scale units
(clean  =                   no) Symmetry clean before centering ?
(rclean =                   6.) Cleaning radius in scale units
(rclip  =                   9.) Clipping radius in scale units
(kclean =                   0.) Rejection limit in sigma
(mkcente=                  yes) Mark the computed center on display ?
(mode   =                   ql)
```

To modify any of the parameters, use the arrow keys to move to the appropriate line. The cursor will appear to the immediate right of the = sign. Typing a value and pressing return will change that parameter. To enter commands like quit or to run the package you must type a colon while the cursor is to the immediate right of any = sign. The cursor will jump down to the bottom near a ":" prompt. From here you can type "g" to execute the package or "q" to quit.

The biggest issue here will be *maxshift*. Its default value is usually 1. This means that when you interactively click on your star in the field, if you click further away than one pixel from where the computer determines the center you will get a "BigShift" error. Changing the value to 2 will simply let the computer calculate the centroid and you will not have to worry so much about where precisely you click.

You can get help on any IRAF package or task (which will include information on, and explanations of, the parameters) by typing '`help` *`package-name (`*`or` *`task-name`*`)`'. Additional information, including a beginners guide to IRAF, can be found at http://iraf.noao.edu.

Return to the main center parameters menu by going to the ":" prompt and typing 'q'. Once back at the main parameters menu, you can activate the package by typing a colon, and then g.

A "blinking donut" will appear in your *ximtool* window. Position the donut over your target star and press space. It should give you confirmation and coordinates for the star in the IRAF window. You can then continue to click on each of the target background stars that you chose. Press q when done. You will repeat this process for the image of the same field at a different epoch. The data is saved in file names identical to the image file with ".ctr.#" (# - some number) attached as suffixes.

4.4 Matchmaker

Now you have two lists of the same stars from two different arbitrary coordinate systems. The next step is to have IRAF pair these stars together. You do not need to do this yourself when you are using *center*. First, however, you will need to clean up your image.ctr.1 files. They will look something like this:

```
   215.814    214.951    0.823   1.960   0.032   0.034          107  BigShift
lhs62_2sec.00000002.NA.148.989   229.991   5    nullfile              0       \
   148.806    231.000    -0.183  1.009   0.032   0.026          107  BigShift
lhs62_2sec.00000002.NA.82.987    158.989   6    nullfile              0       \
   82.564     161.225    -0.423  2.236   0.041   0.025          107  BigShift
lhs62_2sec.00000002.NA.91.987    133.988   7    nullfile              0       \
   92.000     132.000    0.013   -1.988  0.000   0.000          107  BigShift
lhs62_2sec.00000002.NA.145.989   145.989   8    nullfile              0       \
   146.678    147.207    0.689   1.218   0.027   0.034          107  BigShift
```

after a multitude of header comments. Delete the comments up to the first line of star coordinates, then delete every other line because for next IRAF task you will use, *xyxymatch*, the first two columns must be the x coordinates and the y coordinates, respectively. You will need to edit these files in a text editor outside of IRAF. *emacs* is readily available on our Linux systems and is popular. Other editors include *pico* and *vi*. All of these are Unix programs run outside of IRAF. You do not have to delete anything after the first two columns, plus you will need these columns later when you calculate the error. *xyxymatch* will ignore them. You want your files to look like this:

```
215.814    214.951    0.823   1.960   0.032   0.034          107  BigShift
148.806    231.000    -0.183  1.009   0.032   0.026          107  BigShift
82.564     161.225    -0.423  2.236   0.041   0.025          107  BigShift
92.000     132.000    0.013   -1.988  0.000   0.000          107  BigShift
146.678    147.207    0.689   1.218   0.027   0.034          107  BigShift
```

Hopefully, your files will lack the BigShift problems and possibly contain more than five data points. Four should be the **minimum** number of reference stars used to create the transformation.

xyxymatch will take two lists of the matched stars (they need not be in matched order however) and create one matched list. (This task is under the IRAF package *images/immatch*). You will need to feed it a reference list and an input list; these are your two edited .ctr files. It is your decision on which epoch is assigned each of these names, it really makes no difference. Just remember which is which. The output will look something like this:

```
#
# Column definitions
#    Column 1: X reference coordinate
#    Column 2: Y reference coordinate
#    Column 3: X input coordinate
#    Column 4: Y input coordinate
#    Column 5: Reference line number
#    Column 6: Input line number

      215.810       215.400        233.490       158.310      1     5
      148.810       231.000        276.190       103.900      2     4
       82.560       161.220        372.100       119.070      3     3
       92.000       132.000        380.790       145.590      4     2
      146.570       147.540        329.700       168.810      5     1
```

As you may be able to see here, the two lists were input (accidentally) in the opposite order from each other, but *xyxymatch* was able to determine which stars belonged with which. You can confirm the match by inspection of your two images.

4.5 Transformation

After getting the matched list made by *xyxymatch*, the next IRAF task to use is *geomap*. This will create a mathematical solution that will convert one set of coordinates into the other. Again, you will need to edit the parameters of *geomap* (also in *images/immatch*).

This is the edit parameters display for *geomap*.

```
Image Reduction and Analysis Facility
PACKAGE = immatch
   TASK = geomap

input   =                       The input coordinate files
database=                       The output database file
xmin    =                   1.  Minimum x reference coordinate value
xmax    =                 512.  Maximum x reference coordinate value
ymin    =                   1.  Minimum y reference coordinate value
ymax    =                 512.  Maximum y reference coordinate value
(transfo=                     ) The output transform records names
(results=                     ) The optional results summary files
(fitgeom=              general) Fitting geometry
(functio=           polynomial) Surface type
(xxorder=                    2) Order of x fit in x
(xyorder=                    2) Order of x fit in y
(xxterms=                 half) X fit cross terms type
(yxorder=                    2) Order of y fit in x
(yyorder=                    2) Order of y fit in y
(yxterms=                 half) Y fit cross terms type
(reject =                INDEF) Rejection limit in sigma units
```

The input file is file *xyxymatch* created. The database is usually a new file in which you will store your transformation. More than one transformation can be stored in a database file so you will also need a transformation record name next to `(transfo=)`. Be sure to also create the optional summary file and give it a name next to `(results=)`. The min and max values should be as above, as our CCD camera is 512 x 512 pixels. Create the transformation by running the package. Now, look at the summary file that was created. In addition to the parameters of the fit not shown here, it will show you this:

```
# Input Coordinate Listing
#     Column 1: X (reference)
#     Column 2: Y (reference)
#     Column 3: X (input)
#     Column 4: Y (input)
#     Column 5: X (fit)
#     Column 6: Y (fit)
#     Column 7: X (residual)
#     Column 8: Y (residual)

     439.      111.     118.52   375.385   119.8001  376.4461  -1.280083 -1.061066
     494.      140.     57.709   387.853   58.37078  387.5969  -0.661781  0.256073
  486.763    68.742    109.456    439.47   107.8972  438.8269   1.558815 0.6430664
  215.814   214.951    233.495   158.214   232.8828  158.0211  0.6122437 0.1929321
  148.806      231.    276.186   103.903   276.1502  104.2885  0.0357666 -0.385544
   82.564   161.225    372.113   119.213   371.5495  118.0872  0.5634766  1.125793
      92.      132.    380.789   145.585   382.0267  146.7234  -1.237732 -1.138412
```

It shows you the transformed coordinates of the reference list. That is, it transforms the very stars it used to create the transformation and compares them to the input list. So, you can see how close the transformation got. In the above data, it's not that good. Several stars were fit to positions that are over a pixel off. Ideally, every star will be under one pixel.

The residual values in the last two columns are where you can determine the motion of the reference stars. One possible source of error in the residuals is that the reference star itself is moving. If you have high residuals (greater than 0.1 or so) in certain stars it may be wise to eliminate them from the fit.

Note: Remember not to use the high proper motion target star in creating the fit. It should not be present in the file *xyxymatch* creates.

Now that you have created what is hopefully an excellent fit, you can now determine the shift of your target star. Use *center* to ascertain the star's position in both the reference and input coordinate systems (the different epochs). Then, transform the coordinates from one epoch to the other using the IRAF package *geoxytran*.

It may also be helpful to transform not just your target high proper motion star but also another anonymous star in the field that you did not use in creating the transformation. This way, you can see what its "proper motion" is as well.

geoxytran has the same file format requirements as *xyxymatch* so be sure to edit the files center creates to remove any extraneous lines that have the first two columns as non-coordinates. You will need to edit the parameters of *geoxytran* as you did with the other IRAF packages and insert the names of the file containing your target star's coordinates, the file containing your transformation database, and the name of your transformation.

Having transformed your target star, you should get a pair of similar but not identical coordinates in two separate files. Since this star is of high proper motion, you do not want the pixel differences to be zero but appropriate for the time that has passed and for the plate scale of the CCD camera. In a spreadsheet, calculator or your favorite mathematics software package, determine the total proper motion via a simple Pythagorean calculation. You now have the star's total motion in pixels. The next step will be to use the plate scale of the CCD (arcseconds/pixel) to convert this to an angular displacement.

5 Calculating Plate scale

The first task is to correlate background stars on your image with known stars in a catalog. For a visual fit, go to a program such as Starry Night on the computers in the 13th floor lab. Type in the coordinates (RA and dec) for the high proper motion star, and, after the program has centered on the star, zoom in until the square labeled CCD 16" appears. That is the approximate view on the CCD for the 16" telescope. Orient a printout of the observation you made on the telescope with the view on the computer. Then, choosing appropriate, relatively bright stars, move the mouse cursor over the stars on the computer and write down their names and coordinates (here, again, you may need to printout the view from Starry Night to compare it with your actual image).

Another option for looking up the stars is to go to the Simbad online catalog (http://simbad.u-strasbg.fr/). Click on "Query by identifier, coordinates or reference code". Then type in the name of the star in the blank input box (ex: LHS 8), and press "SUBMIT". This should bring up the page for your star. Scroll down to Aladin image and change the radius to a few arcminutes (our CCD produces an image 8 _ minutes on a side). Press "Query and Plot around". This loads a new page listing the catalogued stars within a few arcseconds and their coordinates.

Once you have identified some of the background stars on your image, you can proceed to calculate the CCD plate image scale by finding the distance in pixels between two stars divided by the difference in arcseconds between these two. Open the file that you made with the center package in IRAF for your image. Therein are the coordinates of the background stars in pixels. Use these to find the distance between the stars in pixels.

Now, you must find the distance between the stars in seconds of arc. Taking the coordinates of the stars, find the differences in their declensions and right ascensions as well as the average of the declensions. Let

d = seconds of degree

$\Delta\alpha$ = difference in right ascension

$\Delta\delta$ = difference in declension

δ = average of declensions

then

$$d = 3600\sqrt{(15 \quad \Delta\alpha \quad \cos\delta)^2 + (\Delta\delta)^2} \qquad (1)$$

This will give you the measured difference between the stars in arcseconds. Then, you divide the separation in arcseconds by the difference in pixels, and you have the plate scale of the CCD in arcseconds per pixel. You can do this several times for different combinations of background stars in your image. Average the values you obtain with each pair and this should yield a reasonable plate scale of the CCD.

With the plate scale of the CCD in hand, take the proper motion found in pixels from the IRAF calculations, divide it by the time that passed between the taking of the two images and multiply by the plate scale of the CCD. This is the absolute proper motion of your star in seconds of arc per year. Compare your value against the LHS catalog value for the proper motion of your star, taking into account the time that has passed between the two epochs. (To the day is more than sufficient.) The only thing remaining is to calculate your percent error.

6 Error Estimations

You need to give an error estimate on your derived proper motion value. As the saying goes, "a measurement without an error estimation is not a measurement."

There will be two primary sources of error to consider here. First, there is an error associated with each of your centroid calculations. In the .fit.ctr.1 files, each time you pressed the space bar, the center package created an entry. The first line of the entry that we deleted earlier contained the file name from which the star came and the coordinates of the starting point pixel (where you pressed the space bar). The second (and now only remaining) line of each entry contains the information of greater interest to us: first, the x and y coordinates of the center it found for the star, then the shift in pixels for the x and y direction, the error in the x and y directions and whether there was a major error in calculating the center (i.e. did it find the center outside of the maximum shift value you specified earlier). So, the fifth and sixth columns represent your centroid error.

The second source of error enters during the transformation. When *geomap* created the transformation database that converts coordinates from one image to the other, it considered the shift, rotation, and possible scaling factor necessary to make all your background stars "line up" as best as possible. It does this by attempting to minimize the sum of the squares of the distances between each star's location in the respective images – the so-called "least squares" technique. The square root of this minimized sum (root mean square) is the standard deviation associated with the fit. This gives an indication of error.

If you take a look at the database file created by *geomap* you'll see that it lists all this information for you. The "xrms" and "yrms" values listed are the root mean square errors associated with the transformation. These can be considered as your (x, y) errors arising from the transformation.

Use the standard propagation of errors formula for your calculations. It says that for a function $f = f(x_1,x_2,x_3,\ldots)$ the error associated with f (δ_f) is

$$\delta_f = \frac{\partial f}{\partial x_1}\delta_{x_1} + \frac{\partial f}{\partial x_2}\delta_{x_2} + \frac{\partial f}{\partial x_3}\delta_{x_3} + \ldots \qquad (2)$$

where δ_{x1}, δ_{x2} and δ_{x3} represent the errors on x_1, x_2 and x_3 respectively.

For example, when you calculated the difference in pixels between your target star's position in the new image and that in the transformed old image, you used a Pythagorean distance between two points:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \qquad (3)$$

If we now assume that each of these points has an associated error, utilization of Eq. 1 yields an error for d of

$$\delta_d = \frac{1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}\left[(x_2 - x_1)(\delta_{x_1} + \delta_{x_2}) + (y_2 - y_1)(\delta_{y_1} + \delta_{y_2})\right] \qquad (4)$$

Here, δ_{x1} and δ_{y1} will be the error from the centroid calculation; while δ_{x2} and δ_{y2} will include both the transformation error, as well as the original centroid uncertainty.

Perform a similar calculation to obtain the error in your plate scale. Finally, determine the error bars on your proper motion value by repeating this process again for that derivation.

7 Conclusion

The method we are using here is, admittedly, imperfect. The background stars may have larger proper motions than we are assuming.

In your lab report, be sure to note how accurate your fit was by what was reported in the summary file. Indicate which stars you used to create your fit and where they were on the field. Include printouts of your star images for both epochs, and include the exposure times and dates for both epochs. Explain any large deviations from the value given in the catalog.

The more time that has passed between the two epochs the greater your ability to accurately find the proper motion. This is of course true for anyone attempting to calculate proper motion, regardless of equipment or method.