AN ADVENTURE DERIVING THE SLANT ANGLE OF A SPECTRUM

By
Stanley A. Gorodenski

Free spectroscopic software exists for processing a spectrum. The methodology used by these programs for estimating a slant angle from a .fit image of the spectrum is not readily apparent (to me). This write-up is my adventure for determining a slant angle from an image not as a .fit file, but as a text/data file.

PROCEDURE

My equipment is a Meade 16” LX200R, a LHIRES III spectrograph, an ST-8XME camera and CCDSOFT software for imaging the spectrum, and a ST-i camera and CCDOPS software for guiding. CCDOPS is old imaging software that came free with an SBIG camera purchase. CCDOPS disappeared when the company was taken over by Diffraction Limited, but it can still be downloaded from their website. It has the capability of creating a text/data file of a .fit image.

Before a spectrum can be corrected for slant, an estimate of the slant angle is needed. The method I developed was patterned to a certain extent after the one in IRIS (which predated ISIS).

Using CCDOPS I created a text/data file of a .fit image for SPSS, a statistical analysis program with good programming capabilities (at least for its time). SPSS was used to crop the image of the neon lines, and for estimating the slant angle using linear regression as well as fitting a third order polynomial to interpolate slant angles.

For a long time I could not figure out how to get the slant of a neon line from the cropped text/data file of the .fit image, and I spend considerable time writing programs with different approaches trying to do this. Finally, I saw the light. I am sure most anyone reading this would have immediately seen the solution, which is very simple, but it was difficult for me because I was thinking along a different path.

The solution is this. The cropped text/data file is a Nr X Nc matrix, Nr rows along the Y-axis and Nc columns along the X-axis. For each pixel column along the X axis (from 1 to Nc) the pixel location, called M, on the Y-axis of the maximum pixel intensity is obtained. A linear regression of M against Nc is then executed. The slope of the estimated linear regression line, the predicted values, is the slant angle.

ANALYSIS

Figure 1 is a graph that shows the actual date, i.e., M, the horizontal blue lines, or steps, and the predicted locations from linear regression. It can be seen that the fit between the actual and predicted is very good, but it seemed to be too good because the significance level of the regression coefficient was absurdly high, being out to more than six decimal places (instead of a 0.05 or 0.01 significance, where 0.01 is normally considered highly significant, it was something like 0.000005).

I saw one possible cause for this. The blue horizontal steps in Figure 1, the actual data, consist of, on average, about 18 consecutive pixels along the X-axis, whose maximum intensity values are so close to each other that that they have the same pixel location designation along the Y-axis. You can see there are 16 equally spaced steps that represent the point where the next maximum intensity along the X-axis was large enough to jump to the next step (pixel location) on the Y-axis. The slant of a spectrum is usually so small, around 3-4 degrees, that for the ST8-XME camera (with a 1020 x 1530 chip) it takes about 18 pixels along the horizontal axis before maximum intensities jump to the next pixel step along the Y-axis.

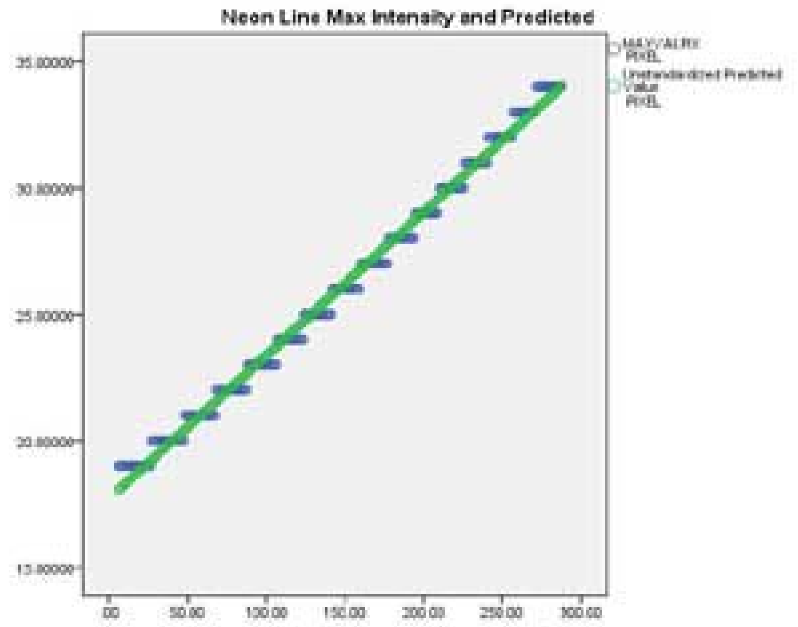

Figure 1. Actual and predicted values of the neon line at pixel location 321.

These horizontal steps consisting of 18 data points each with identical Y-axis location values, pixels, could be interpreted as duplicate data that overly inflate the significance of the linear regression coefficient. However, there is a gradual change in maximum intensity value among the 18 pixel location but this is hidden because the metric is a pixel.

Assuming they do represent duplicate data, for each step I used SPSS to compute the average of the pixels on the X-axis that had identical consecutive Y-axis M variable values and assigned the M value to it. I call this the Collapsed data. The regression result of the Collapsed data is graphically shown in Figure 2.

The significance level of the regression coefficient was still absurdly high, again out to more than six decimal places. However, the actual and predicted values almost completely overlap (in Figure 2) and so this high significance level may be correct. I used the regression results from the Collapsed data to estimate slant angles.

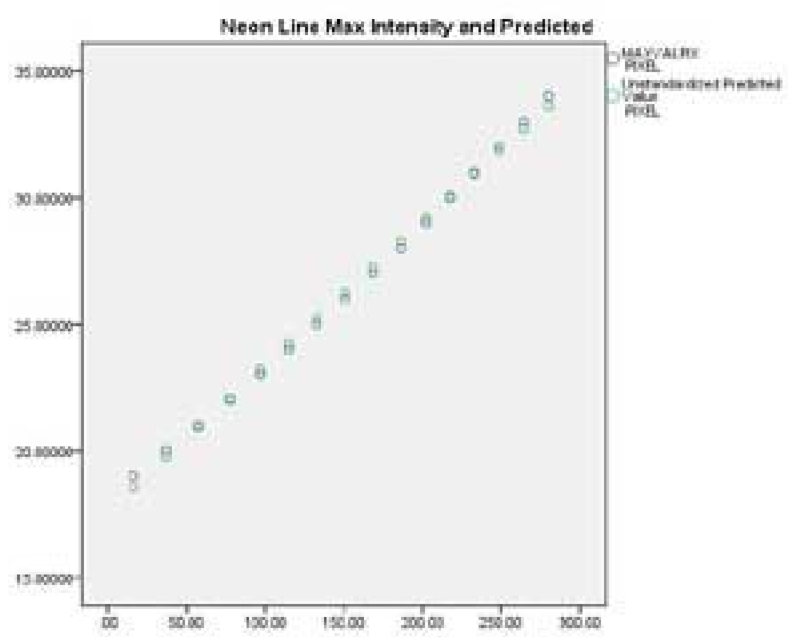

Figure 2. Actual and Predicted values of the Collapsed data from linear regression. This is a graph of one of the four neon lines, the line at position 321, in the region of the Sodium D1-D2 region. The other three lines have nearly identical graphs and so they are not shown.

The neon frame for this article are in the Sodium D1-D2 region and consist of four neon lines. The pixel locations of the lines and the wavelengths they represent are in Table 1.

Pixel Location	Line Wavelength
321	5852.5
541	5881.9
1019	5844.8
1256	5975.5

Table 1. Pixel locations and associated wavelengths of the neon lines within the Sodium D1-D2 region.

A neon frame is always taken just before the integration of the target spectrum and one immediately after. The frames are called Neon1 and Neon2, respectively. The computer program I wrote computes all the slant angles for the four lines in each frame in one execution of the program, as well as all the other graphs and data in the rest of this write-up. Table 2 is the computed slant values for each frame along with the corresponding estimates from ISIS.

Table 2.

PIXEL	REGRESSION	ISIS	!	REGRESSON	ISIS
LOCATION	NEON1	NEON1	!	NEON2	NEON2
321	3.27	3.31	!	3.32	3.32
541	3.36	3.35	!	3.38	3.36
1019	3.42	3.46	!	3.48	3.47
1257	3.50	3.50	!	3.55	3.50

As can be seen, the agreement between my regression methodology and that of ISIS is very good, differing by only at most 0.04 in three instances.

Once the slant values are computed, the program then averages the two slant values from Neon1 and Neon2 for each of the four pixel locations. A third order polynomial model is executed that produces predicted slant values for each pixel location along the X-axis. Figure 3 shows the shape of the curve after applying a spline interpolation.

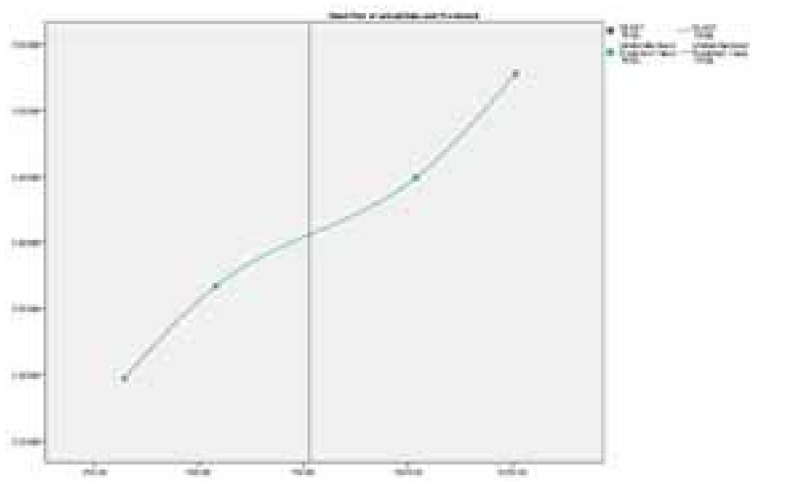

Figure 3. Third order polynomial of the predicted slant values.

Because a cubic regression is applied to only four data points, the regression fits the data perfectly and so the predicted and actual values are equal irregardless of any significance criteria. In spite of this, I feel confident in using the interpolated predicted values because the differences in slant values appear to be more the result of systematic trends in the image instead of random variation.

The program also lists the predicted slant values in intervals of 50 from 1 to 1530 pixels (1530 pixels is the number of horizontal pixels in the image of the ST-8XME camera). If I am interested in a particular spectral line or region I can enter the pixel location at the start of the program (instead of wading through syntax to get to the right place) and this slant value is part of the output along with the list of interpolated values. If one is interested in getting a slant value that extends beyond the beginning and ending data points in the graph, it probably would be best to use the slant value of these ending points instead of extending the predictions beyond them.

CONCLUSION

The program appears to do very well in estimating slant angles. The slant values can range from 3.27 to 3.55 (Table 2) among the four neon lines. Consequently, choosing only one line for a slant could be risky. Once the cropping parameters and a few other parameters are defined, one run of the program quickly produces the results in Table 2 (omitting the ISIS results), Figure 3, and a list of interpolated values. The interpolated values are an advantage in computing equivalent width and can be applied to either the center of the spectral range or any other place in the image. However, this kind of accuracy may not be necessary and probably amounts to nit picking and so the quick way in ISIS is probably the best way to go. Irregardless, this was a (almost)fun project.