**The Fun of Processing a Stellar Spectrum - the HARD WAY**

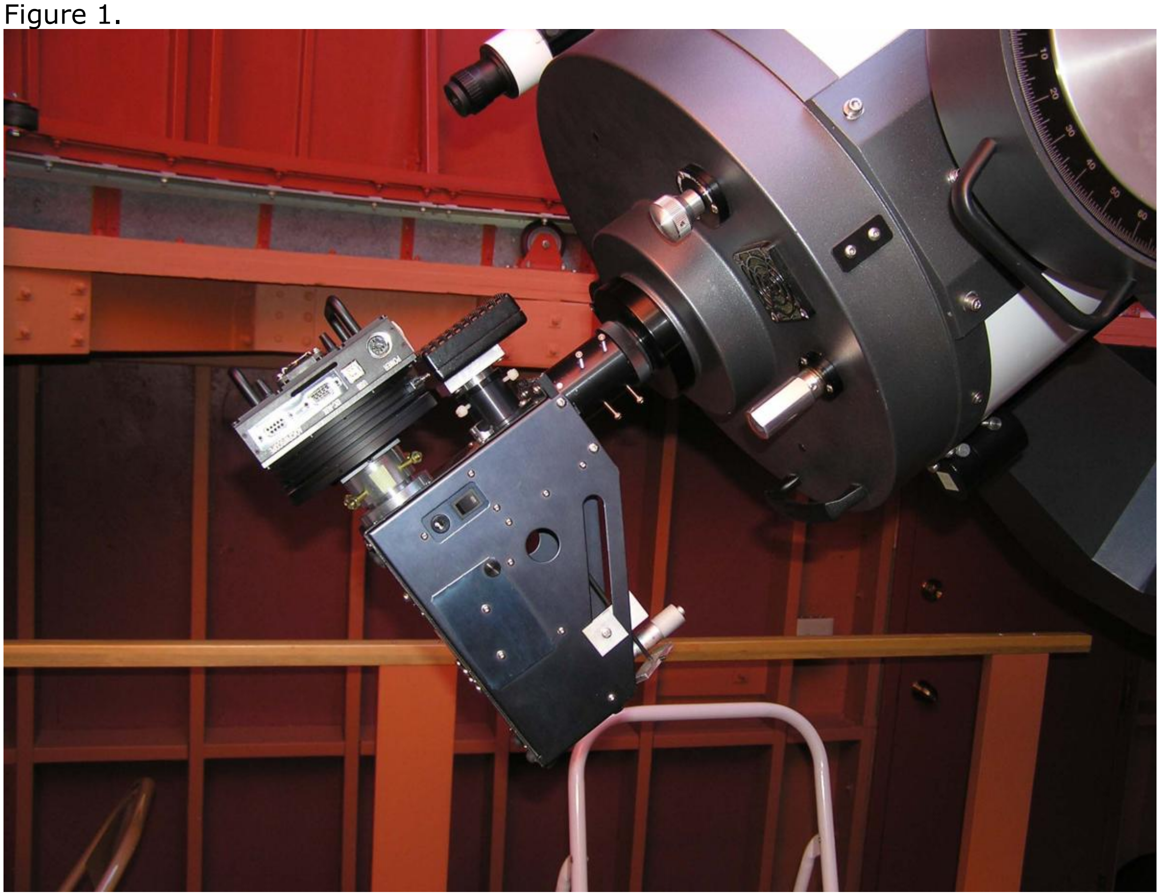
by

Stanley A. Gorodenski

Presented at the November 2018 AAVSO meeting in Flagstaff, Arizona

I live in Dewey, about 70 miles south of Flagstaff. My setup for spectroscopy is a 16" Meade LX200R telescope, an LHIRES III spectrograph, and an SBIG ST-8XME camera.

Figure 1.



A raw spectrum has to be processed so it can be compared to spectra taken at other times, locations, and with different equipment. Figure 2 shows some of the freeware that can do it.

Figure 2.

Freeware:

- VSPEC
- IRIS
- ISIS
- AUDELA
- BASS
- MIDAS
- IRAF

RSPEC is not free but the cost is not prohibitive.

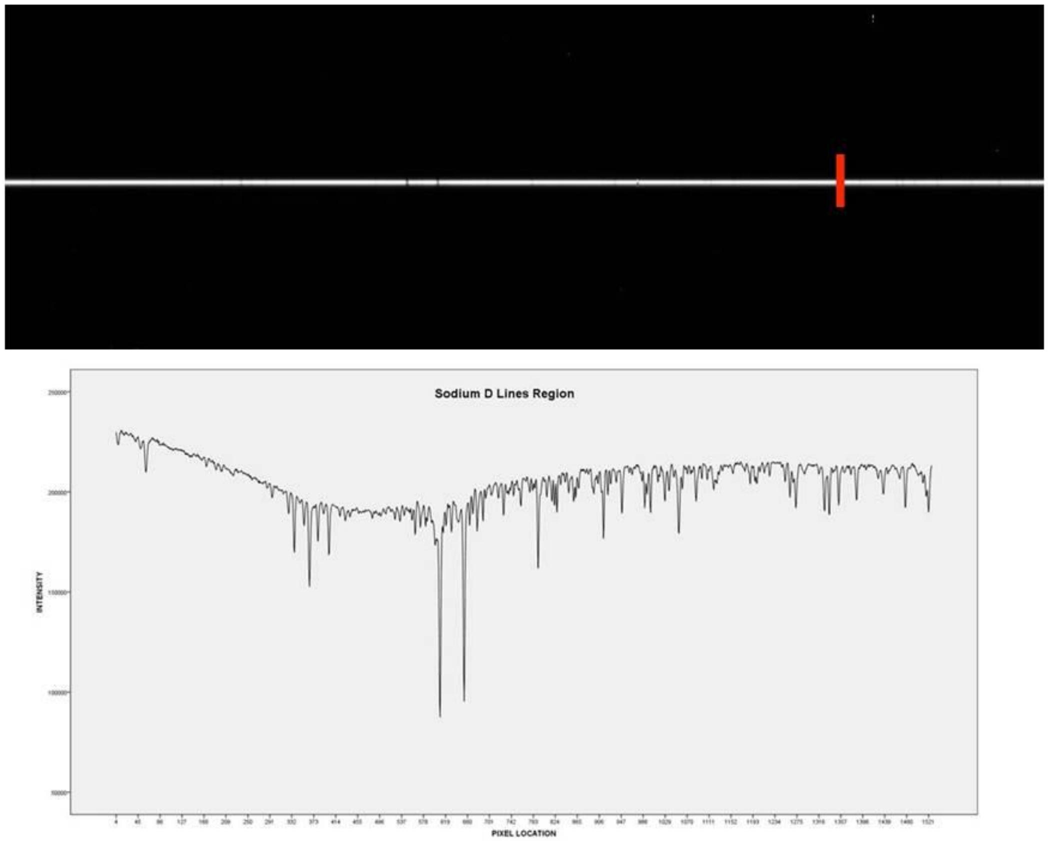
To a degree, these packages are like a black box. As a result, I have written computer programs to do some of the things they do. One purpose of this talk is to show that if I can do it so can you. Plus, what is more interesting, with your own programs you can explore your spectra statistically and graphically to a much greater degree than you can with freeware. For processing, I am not recommending any of the things I will present. They are all exploratory, especially the method for a dark sky subtraction which is still in progress. Instead, I recommend the freeware. In this respect, I find VSPEC to be very good and easy to use.

I learned some computer programming on the job using SPSSX, a statistical package with good programming capabilities. I have used it to estimate continuums, to compute radial velocities, equivalent widths, tilt angles, and calibration models, but for this talk I will first describe a method to clean up a spectrum, and then will describe a method for doing a dark sky subtraction.

## **CLEANING UP A SPECTRUM**

The top image in Figure 3 is what a raw spectrum looks like, a bright horizontal line on a black background. A graph of it, the bottom graph of Figure 3, is called a profile.

Figure 3.

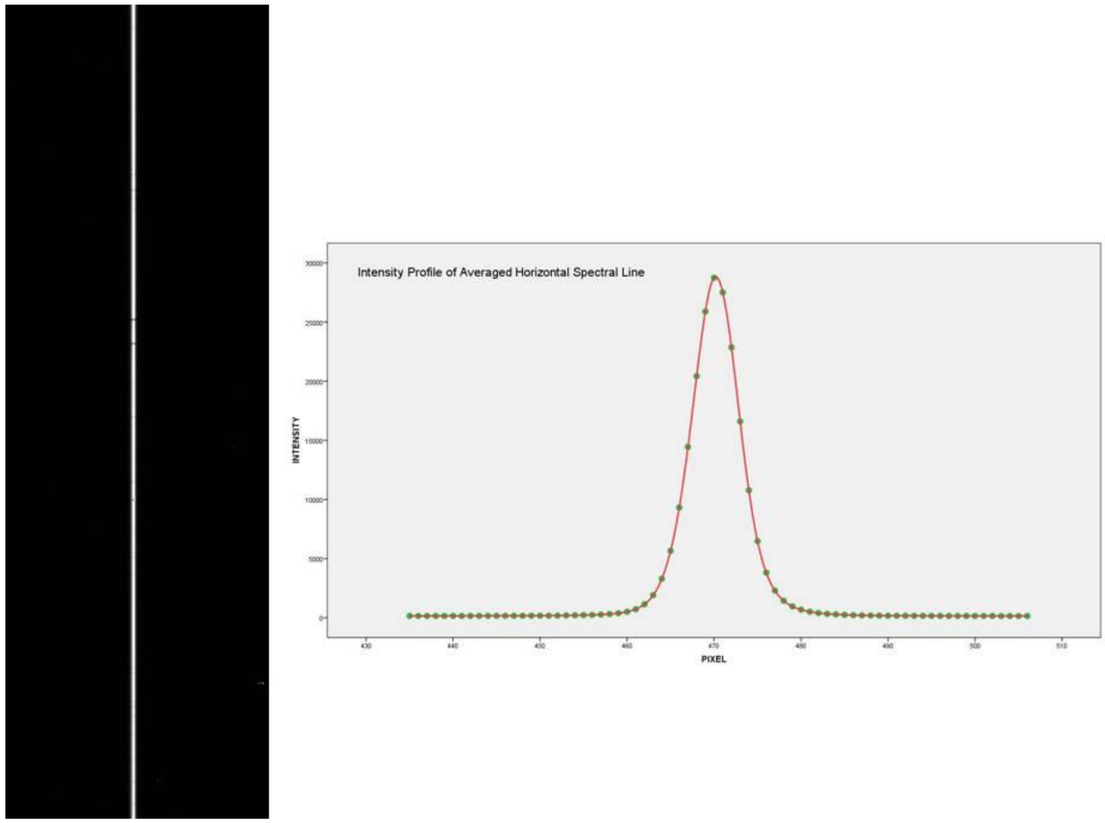



The red vertical line in the top image of Figure 3 represents an exaggerated one pixel wide slice through the spectrum. In this talk I call this one pixel wide slice a pixel column. A graph of its intensities, a profile, should be Gaussian in shape, or close to it. Identifying and correcting column profiles that deviate significantly from a true Gaussian curve is what I call cleaning up a spectrum. To do this a true Gaussian curve is needed as a template.

The first hurdle to overcome to process a spectrum is getting an image in a format so your computer program can read it. Probably not many are aware that CCDOPS, an old SBIG imaging software, can write out a FIT file as a text file.

The true Gaussian curve is constructed from the spectrum itself. The computer program rotates the image 90 degrees, like the image to the left in Figure 4, and then the average of each image column is plotted against pixel location as seen in the graph to the right in Figure 4.

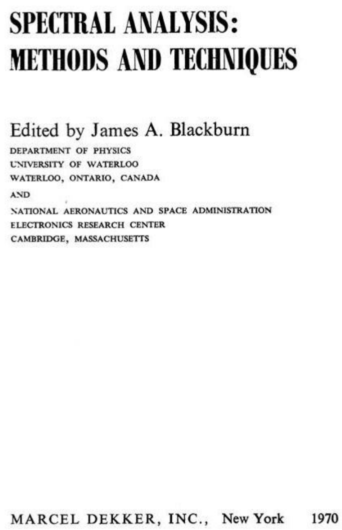
Figure 4.




As you can see, the curve is very Gaussian in appearance.

To construct a true Gaussian curve from it I needed the algorithm to do it which I found in a used book I bought about 20 years ago, or so, maybe even 30, called Spectral Analysis: Methods and Techniques (Figure 5).

Figure 5.



**SPECTRAL ANALYSIS:**
**METHODS AND TECHNIQUES**

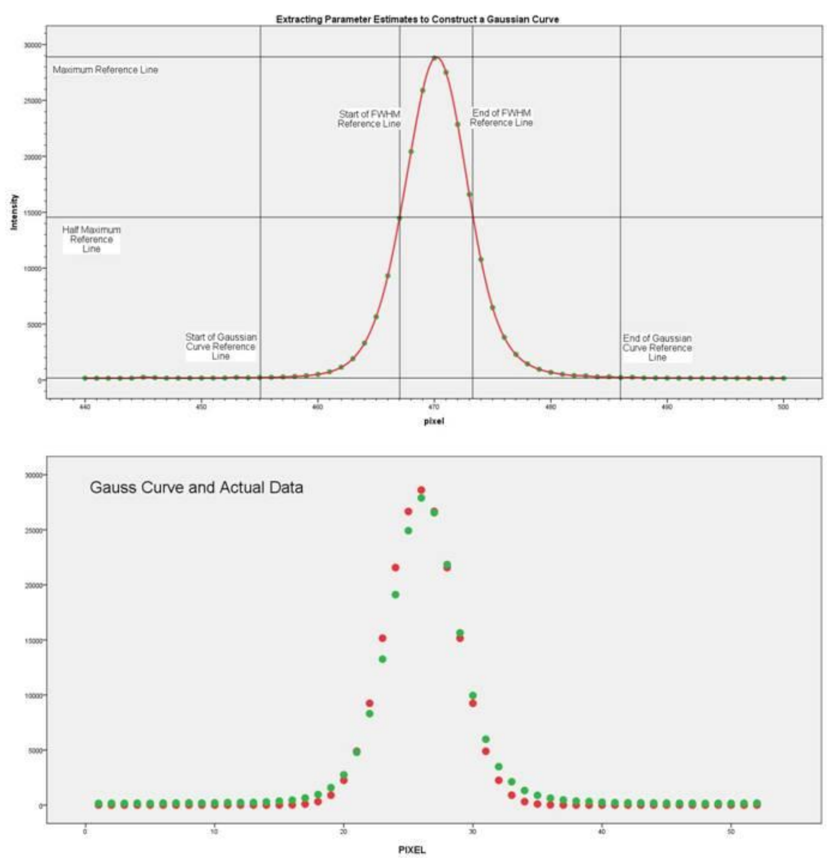
Edited by James A. Blackburn

DEPARTMENT OF PHYSICS
UNIVERSITY OF WATERLOO
WATERLOO, ONTARIO, CANADA
AND
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
ELECTRONICS RESEARCH CENTER
CAMBRIDGE, MASSACHUSETTS

MARCEL DEKKER, INC., New York 1970

A true Gaussian curve is defined by three parameters, the height, the width at half maximum, and the range. These parameters were estimated from the previous curve of column averages, which is the top graph in Figure 6.

Figure 6.



After the true Gaussian curve is constructed the next step is to graphically compare the column profiles to it. The ccd image is 1530 columns wide and so there are 1530 graphs. All the graphs are generated in one run of the computer program. After quickly looking through some of the 1530 it became apparent that almost all are very close to being a true Gaussian as shown in the bottom graph (Figure 6) of one of the column profiles. The red dots are the Gaussian estimates and the green is actual data.

It would be very tedious to manually go through 1530 graphs to identify potentially bad column profiles. To make the computer do it for me I needed some measure of deviation from a true Gaussian. The measure I chose is the definition of the standard deviation (top formula of Figure 7) where the squared term is the difference between the actual intensity and the Gaussian estimate. It worked very well.

Figure 7.

$$\text{Standard Deviation} = \sqrt{\Sigma (A - G)^2/DF}$$

A = Actual Intensity
G = True Gauss Intensity
DF = Degrees of Freedom

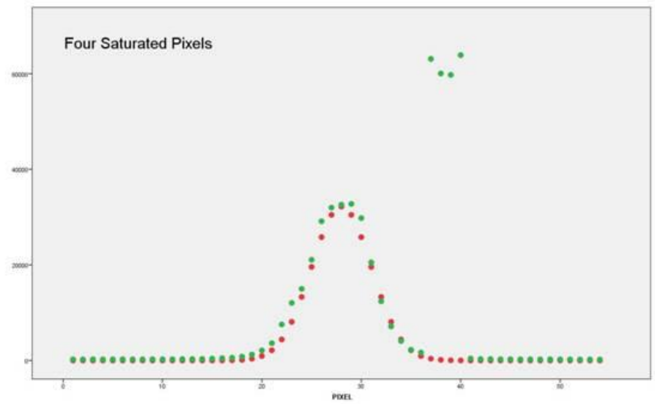
| PIXEL LOCATION | INDIVIDUAL STANDARD DEVIATION | AVERAGE STANDARD DEVIATION |
|---|---|---|
| 117 | 9121 | 828 |
| 237 | 9082 | 828 |
| 582 | 23997 | 828 |
| 607 | 2923 | 828 |
| 608 | 4536 | 828 |
| 609 | 4877 | 828 |
| 610 | 3543 | 828 |
| 653 | 3981 | 828 |
| 654 | 4523 | 828 |
| 655 | 3242 | 828 |
| 680 | 9089 | 828 |
| 931 | 10851 | 828 |

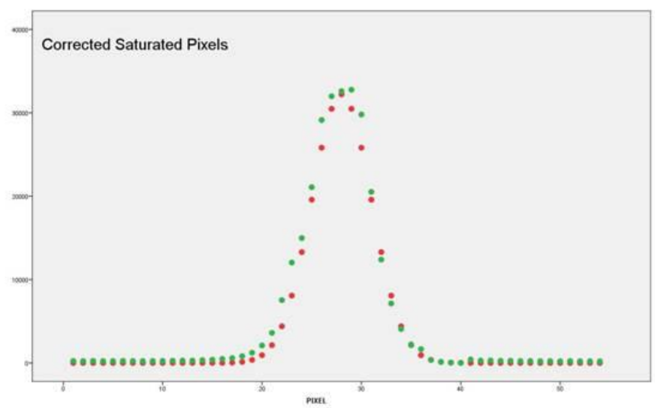
The computed standard deviations of all 1530 column profiles are averaged and then multiplied by the number three, a value associated with the 0.01 level of significance. A column profile whose standard deviation exceeds this value is printed out (the bottom table of Figure 7) and graphed for a visual inspection to see if it actually is a bad profile. Out of 1530 profiles, only 12 had standard deviations that exceeded the significance level.

Originally, the purpose of constructing a true Gaussian curve was to identify potentially bad profiles, but I later realized I could use it to correct them. Here are some examples of how nicely it worked.

In the top graph of Figure 8 four pixels are near or at saturation. This is easy to correct with a programming statement that says if an intensity is greater then 55,000 than it is equal to the value of the Gaussian estimate. The bottom graph of Figure 8 shows how nicely it corrected the profile.

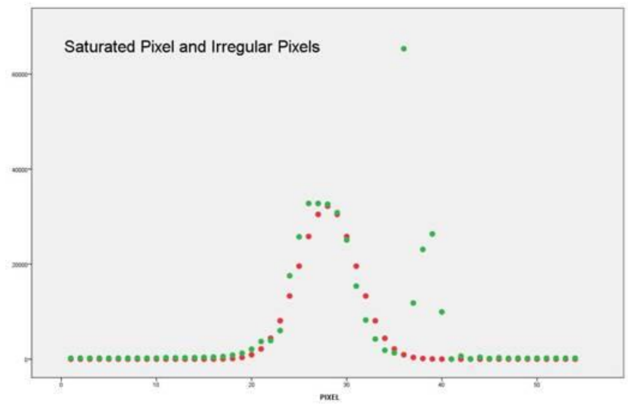
Figure 8.

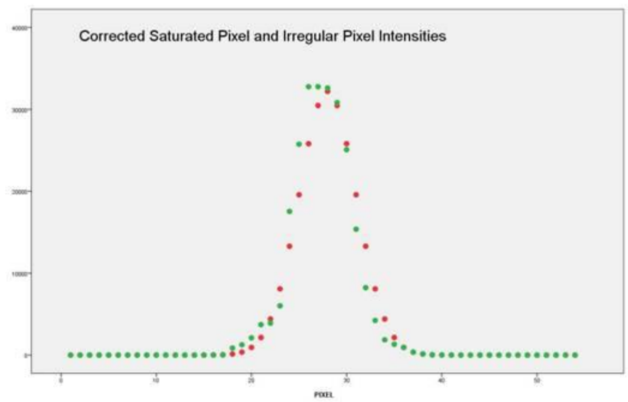



The top profile in Figure 9 could potentially be more difficult to correct because four of the problem intensities in the top graph are within the intensity range of the spectrum itself, but it was easy. I arbitrarily chose the number 10 and put in a statement that says if an intensity is greater than 10X the true Gaussian value it is equal to the Gaussian value. The bottom graph of Figure 9 shows how nicely it corrected the profile.

Figure 9.

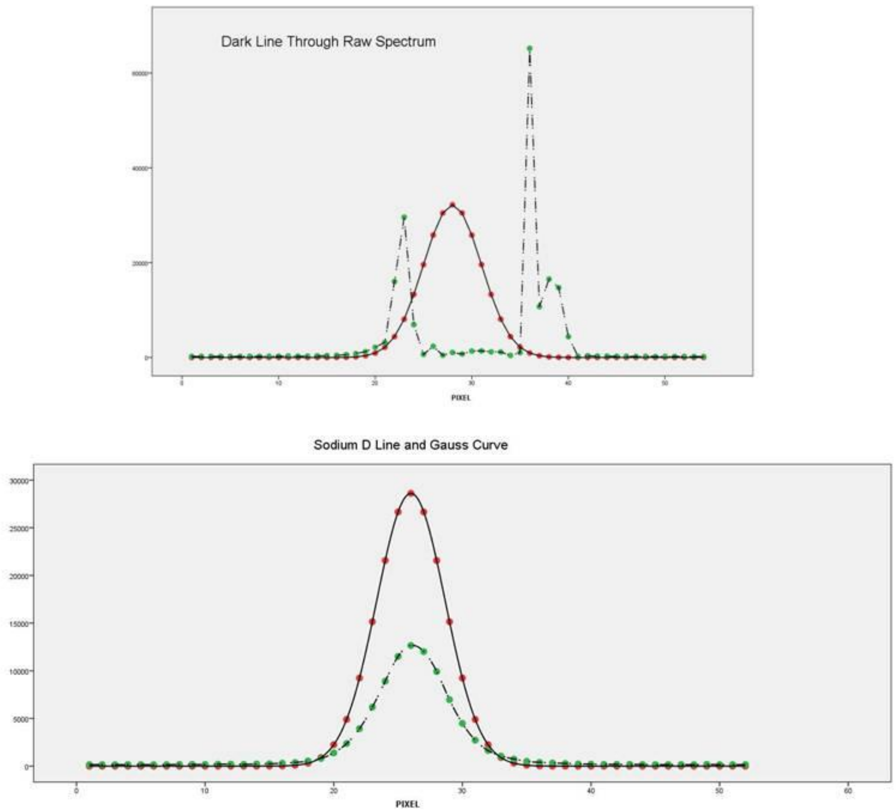



The top profile in Figure 10 is the result of a black vertical line through the raw spectrum. This cannot be corrected like the others. The only choice is to average the column before and the column after.

The bottom profile in Figure 10 was flagged as highly significant, but it is the minimum of one of the Sodium D lines. This is not a problem profile. It was significant only because the peak intensity is much lower than the average.

Figure 10.



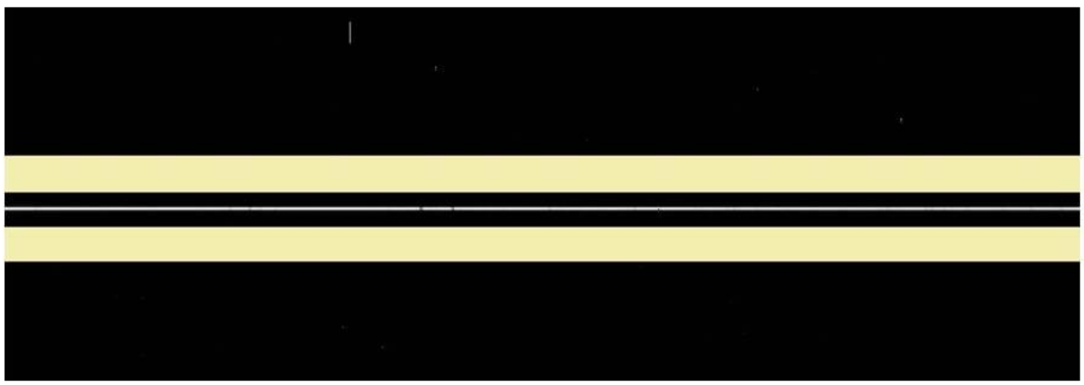

After all the irregular column profiles are corrected, the next step is a dark sky subtraction.

## **DARK SKY SUBTRACT**

The dark sky background, a low level illumination, has to be subtracted from the spectrum, for reasons I won't go into here. My method for a dark sky subtraction could be viewed as an overkill, but with the power of today's computers it is easy to do overkills. It is based on how IRIS, one of the freeware I displayed earlier, does a subtraction, but with, hopefully, some improvements.

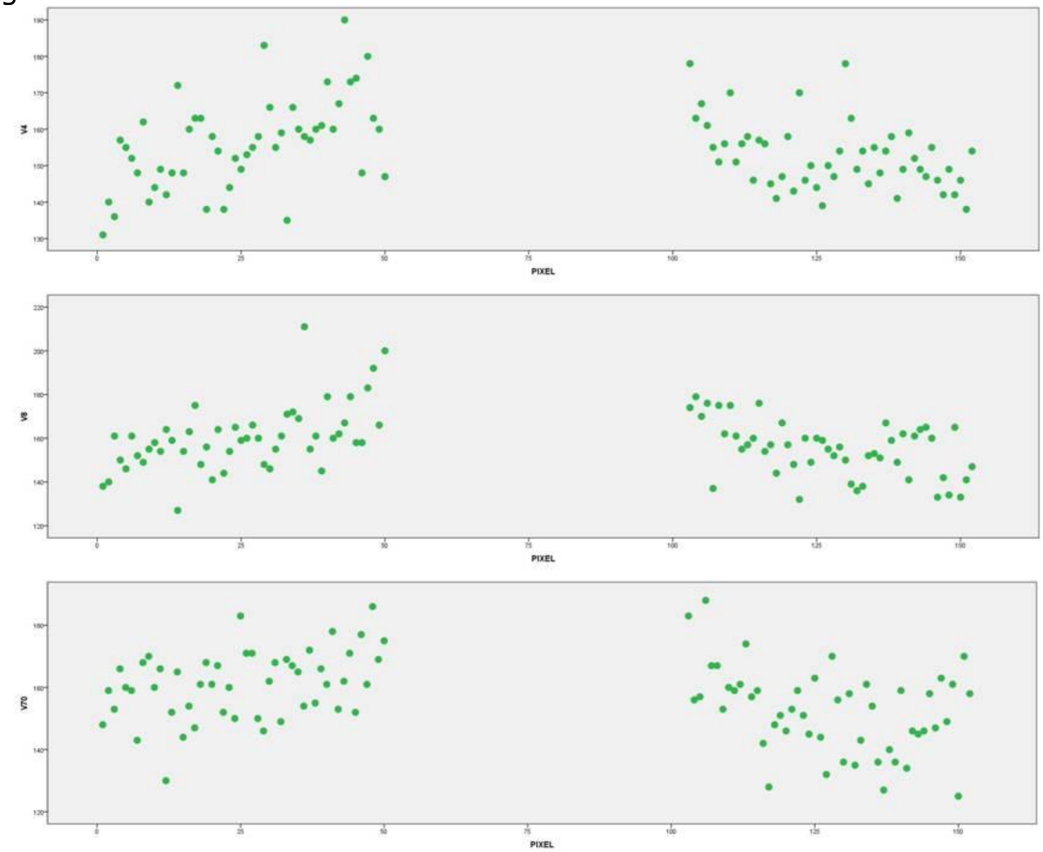
Recall, the top image in Figure 11 is what a raw spectrum looks like. A band, 50 pixels wide, on each side of it is selected as shown in the bottom image of Figure 11. The pixel intensities within these bands are the data for building regression models that estimate the dark sky background intensities to subtract from the spectrum.

Figure 11.



In Figure 12 are three pixel column scatterplots of the two bands. The gap in the middle of each graph is where the spectrum would be. You can see intensities that look atypical, that is they do not appear to follow the general variation of the scatterplot. They need to be removed because they could adversely influence the regression model estimates of the dark sky background.

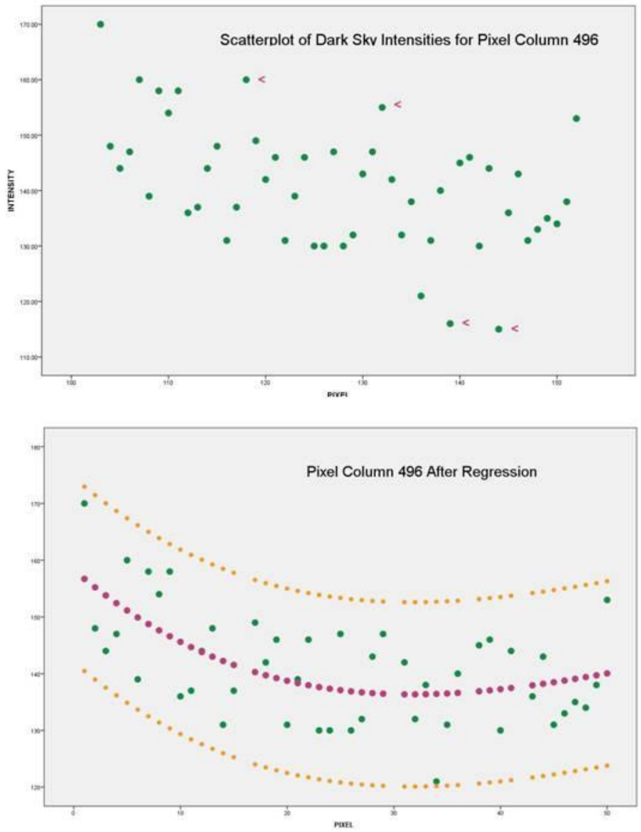
Figure 12.




Each band is processed separately to identify and remove these intensities. To do this the program builds a polynomial regression model for each pixel column. A polynomial model is where the independent variables are the pixel locations raised to exponential powers. Because the image is 1530 columns wide and each column is a variable, it builds 1530 polynomial models up to a third order depending on the significance of the coefficients. A third order is a pixel location raised to the third power. The routine I wrote also computes the 95% upper and lower confidence levels, and sets those intensities that exceed these levels to system missing. Setting to system missing is the way to remove these atypical intensities. All this is accomplished in one run of the computer program.

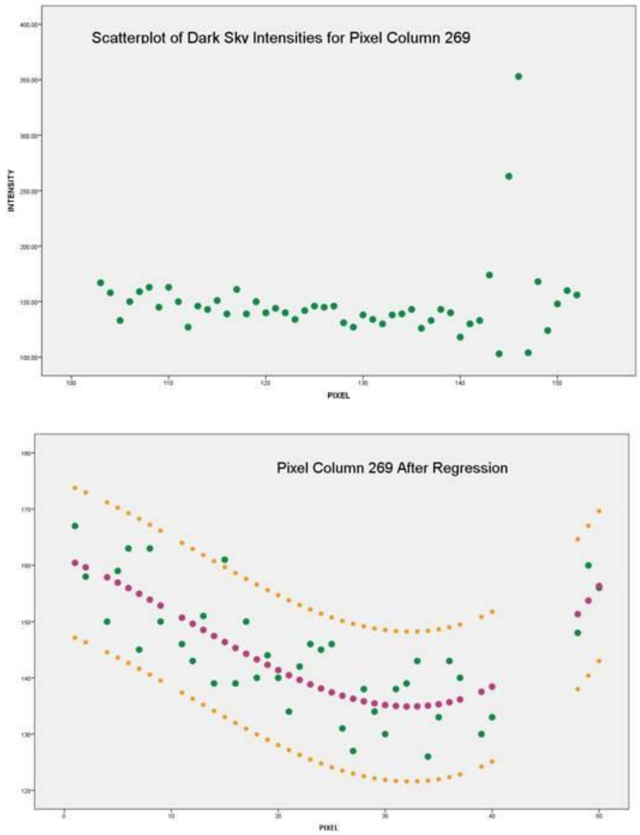
Here are two examples of how this procedure worked. The top graph in Figure 13 is pixel column 496. You can see four intensities that are not that extreme, but they look atypical. The red arrows point to them. The bottom graph of Figure 13 shows the program set them to system missing because they exceeded either the 95% upper or the 95% lower confidence level. The four gaps in the graph are where they would be. Also displayed are the upper and lower confidence levels and the fitted regression line.

Figure 13.




The top graph in Figure 14 is pixel column 269. There are some extreme intensities plus some that are less extreme but they still look atypical. The large gap in the bottom graph of Figure 14 shows the program had set them all to system missing.

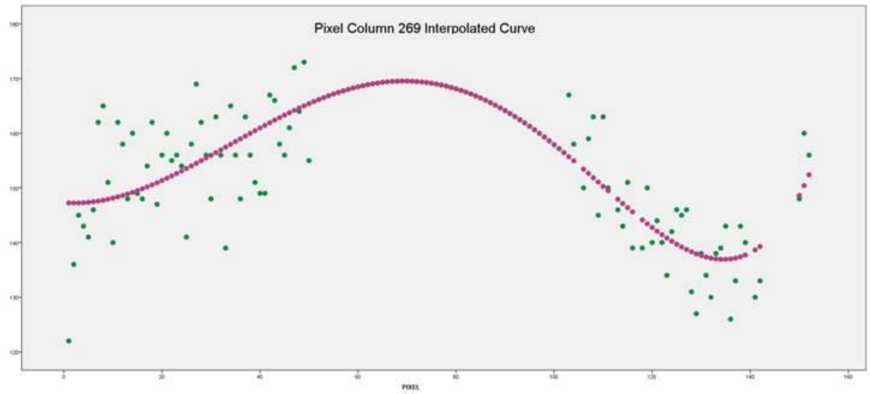
Figure 14.

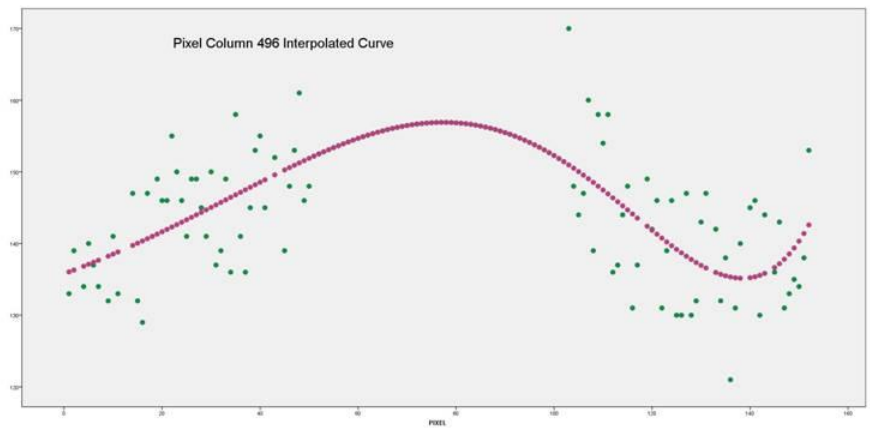



After both bands are processed to remove atypical intensities they are reconstituted to the positions they had occupied in relation to the spectrum. The program then builds 1530 polynomial models, up to a 6th order, using the intensities in the bands on both sides of the spectrum. The models produce the interpolated dark sky intensities as shown in these two graphs.

The top graph in Figure 15 is pixel column 269 and the bottom one is pixel column 496. The gaps in the fitted regression lines are the atypical intensities the program had previously set to system missing. The interpolated regression line between the two clusters of data points in each graph are the interpolated dark sky values that will be subtracted from the spectrum.

Figure 15.

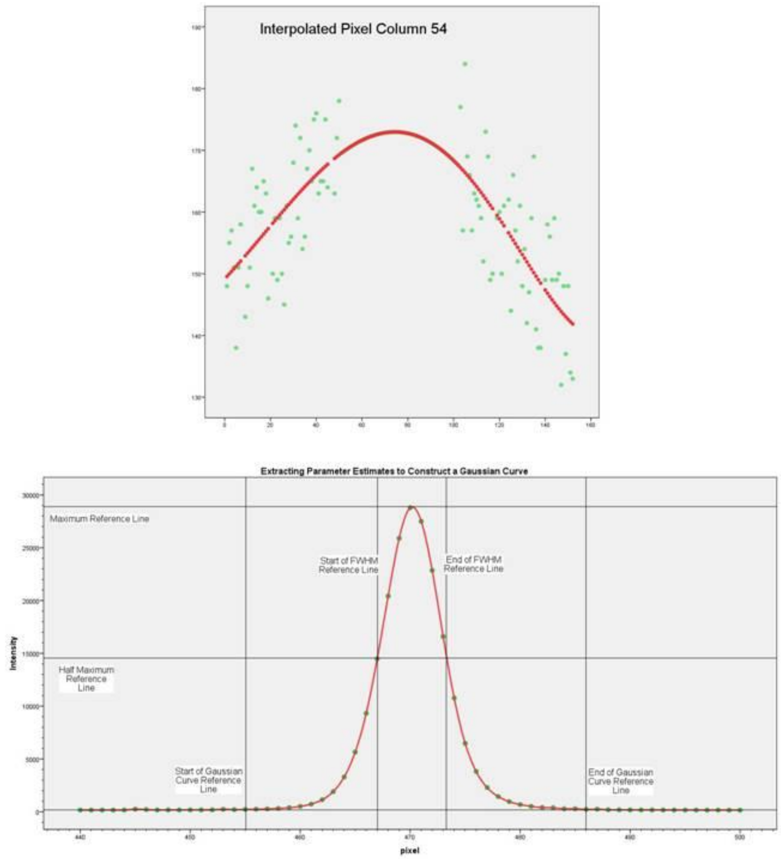



The fit of the regression lines in these graphs appear to be good but not quite as good as they could be. This becomes more apparent when a graph is not stretched horizontally as much as these are. When this is done, it can be seen in the top graph of Figure 16 for pixel column 54 that although the fit does not appear to be too bad, it is not following the trend of the scatterplot. The fitted line should be following the trend of the scatterplot, but instead it is cutting slightly across it.

To understand why it is doing this, recall the bottom graph of Figure 16 was used to estimate the parameters to construct a true Gaussian curve. From this graph I had determined the range of the curve to be from 455 to 486 pixels.

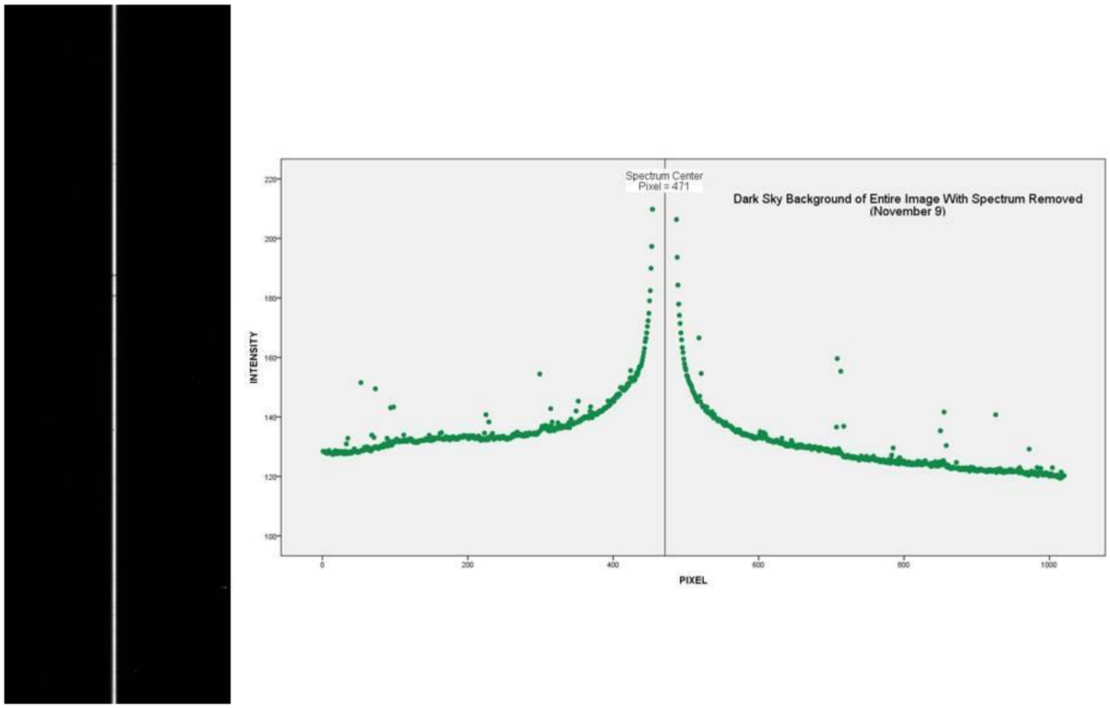
Figure 16.




The curve levels out at the base but this is entirely a graphic scale effect caused by the peak intensity of the spectrum being almost 30,000 intensity units high compared to the sky background which is only in the hundreds.

To see this, remember, the image to the left in Figure 17 is where the graph came from. When the column averages are plotted across the entire image, not a narrow range to estimate Gaussian parameters, you can see in the scatterplot in Figure 17 that the intensities continue to decline in a Gaussian manner, although the steepness of the decline is exaggerated due to a graphic scale effect. It drops over a range of about 155 pixels with an average intensity drop per pixel of about a half of an intensity unit, a very small change compared to the peak intensity of the spectrum which is almost 30,000 intensity units high. The spectrum within the 455 to 486 pixel range was removed. This made it possible to see that the spectrum continues to drop in intensity in a range of over a hundred pixels beyond what had initially been determined to be the end points of the curve, the 455 and 486 end points.

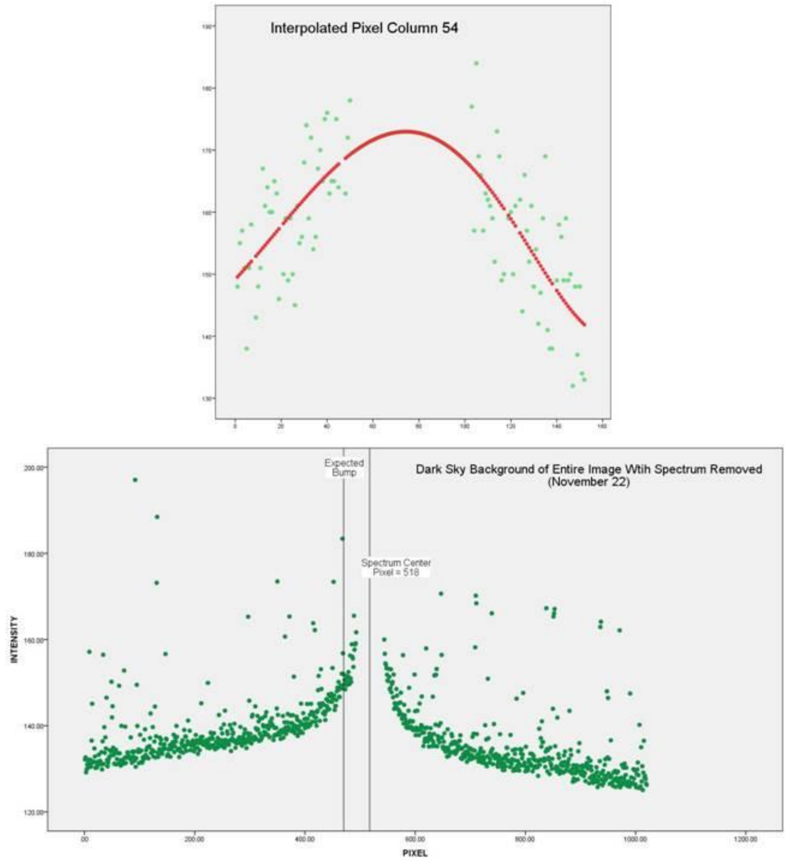
Figure 17.




It is now obvious the trend of the scatterplot for pixel column 54 in the top graph of Figure 18 is the trend of the spectrum declining in intensity. The bands for building the regression models were in these extended ranges and, therefore, includes the spectrum. I do not think the spectrum being in the bands necessarily creates a problem for a dark sky subtraction because when the interpolated dark sky values are subtracted from the spectrum the result would be equivalent to a slightly shorter imaging exposure time.

Second, I think subtracting the interpolated values from the spectrum should still give good results if the regression procedure was applied consistently across all pixel columns, although I am not certain of this. The ideal solution to a regression model that does not follow the trend of the scatterplot of intensities is to place the bands outside the extended ranges of the spectrum. It seems the problem with this solution is that 140 or more pixels is a pretty long distance and setting the bands this far out may get into localized irregularities in the dark sky background or image. Maybe the best one can do is develop a method that is consistent in the sense that if the intensities of a pixel column are lower than those of another pixel column, the interpolated dark sky values will be proportionately reduced in intensities. I have no idea if the method described here is consistent.

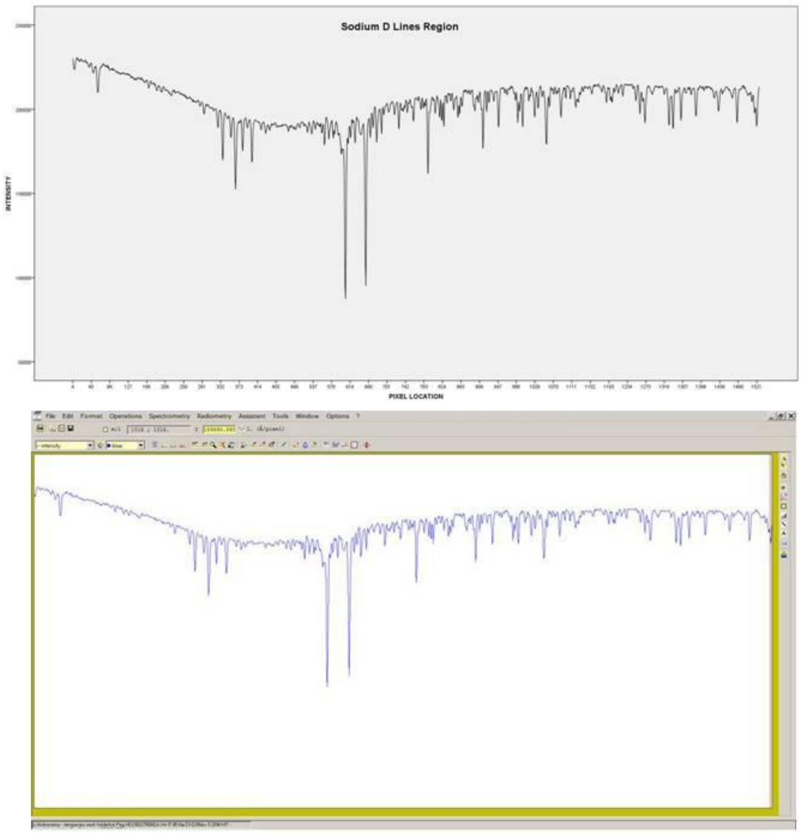
Figure 18.




The bottom graph in Figure 18 is a scatterplot of another spectrum. Notice that the intensities also continue to drop to both sides of the image. However, the drop, just like in the previous spectrum, is relatively linear after the Gaussian part of the scatterplot. The explanation for this is that the CCD image was not flatted. The further away from the slit, the dimmer the light source becomes. If This explanation is correct, intensity declines should be able to be corrected with a good flat applied, which is another argument in favor of flats.

Getting back to the procedure for subtracting the dark sky. After the dark sky is subtracted, the program then displays the entire profile of the spectrum for me to look at. If nothing in it looks odd, like a single pixel spike, I let the program proceed to the end where it writes out the profile as a .dat file to finish the processing in VSPEC. The top image in Figure 19 is the profile as it appears in SPSSX output, and the bottom is what it looks like from the .dat file in VSPEC.

Figure 19.




When I first created a profile of a spectrum, it was a thrill to have been able to do it. At that time I chose the maximum intensity value of a pixel column as the intensity value for the profile. This is wrong because it can lead to some strange looking profiles. Besides this, I do not think a profile based on just one value, the maximum intensity, is sound from a statistical point of view. What I do now is use the addition of all the intensity values of a pixel column. Of course, pixel columns cannot contain missing intensity values. Hence, the need to replace a bad pixel intensity with some kind of estimate, like the Gaussian replacement method I described earlier, instead of setting it to system missing.