Weekly Report

John Anglo

June 1, 2012

1 Working with MSP data

1.1 Checking the integrity of data

Looking at the photometer data correctly requires being able to determine when the struct arrays created by `norstar_msp_readfile.pro` do not represent contiguous and complete series of counts. The struct array contains a field called `mirror_step`, evidently meant to represent the photometer's position along its north-south sweep, that increments over the period of one sweep and resets for the next sweep. I used this field to check the integrity of the arrays, when combining hour-long arrays to form whole nights' worth of data, before turning the data into 2D plots.

The data collected from the first 10 days of November 2011 by the Fort Smith photometer proved to be convenient for this kind of work, as during this period each data set began at hour UT00 and ended within the day itself. This way, the file search performed by `norstar_msp_readfile.pro`, which searched for files within a day, could be expected to produce a complete and well-behaved set of data without modification.

1.2 Identifying stars

I attempted to compare what were identified as stars in the 2D plots with views of the sky from the photometer's approximate coordinates, as given by JSkyCalc. The struct array also contains the field `seconds`, which gives the time at which a count was taken, in Unix epoch time. This meant I could extract the time at which a star passed the photometer's field of view, give that time to JSkyCalc and determine whether a star passed the meridian at that time.

There is a recurring period in the November data during which a group of stars can be clearly seen passing through the field of view. Using JSkyCalc I identified two of the stars as Vega and Deneb, as they appeared brighter than the rest and passed at about the right time. I was not able to determine whether the altitudes of each star were as expected, because we don't yet know how to find the pointing angle of the photometer from the data. It cannot be assumed that `mirror_step` corresponds linearly to the pointing angle; the 2D plots show that stars are vertically narrower in the middle than closer to the top and bottom of the plot, which correspond to north and south respectively. This suggests that stars spend less time in the field of view of the photometer in the middle of the sweep, which implies that the photometer is accelerating, then decelerating again during its sweep.

To provide further show that the "stars" we found were stars, I tried to show that because the length of a sidereal day is twenty-four hours minus about four minutes, looking at a region of the plots that contained stars, starting at the same time each night, one could see stars moving leftward, crossing the field of view earlier than the night before. I was able to confirm that the same group of stars that possibly included Vega and Deneb could be seen in the first 7 days of February 2011 and traveled to the left by about four minutes a day.

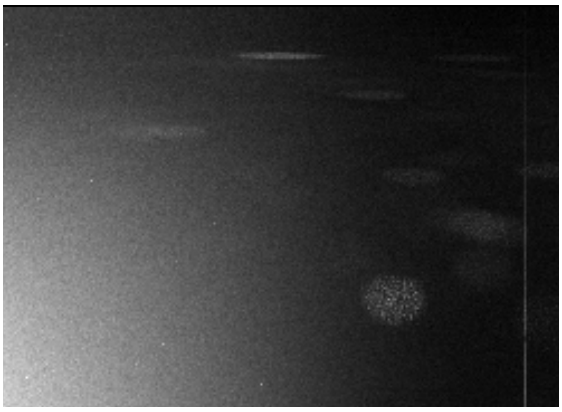

Figure 1: Stars as seen by the photometer on November 6, 2011. Note that the bright star in the southern end of the image, identfied to be Vega, is round, while the one to the north, possibly Deneb, is a horizontal sliver in comparison.

2 Working with imager data

I started by attempting to recreate the keograms included with each day's collected data. This required not only identifying gaps like those in the MSP data, but also incorporating them into the data, to produce an image that accurately represented the time at which a column of data was collected by the imager, with its horizontal position on the image. I did not know of any field stored in the metadata, produced by the `themis_imager_readfile`function, that could be used similar to `mirror_step` in the MSP data, so instead I used the `exposure_start_cdf` field, which stores time like `seconds` in the MSP data. Knowing that the `stream2` set contained images collected every minute, I wrote a program that would compare the time on one image was taken to that of the previous image, and if there was a gap larger than one minute, it would fill the gap with as many "frames" containing zeroes as necessary. After a few trials on different sets of imager data, I figured this was a reasonable method of filling gaps, as I was able to consistently produce 1440 frames representing twenty-four hours, and performing a similar filling procedure on the time field itself yields a linear plot as expected.

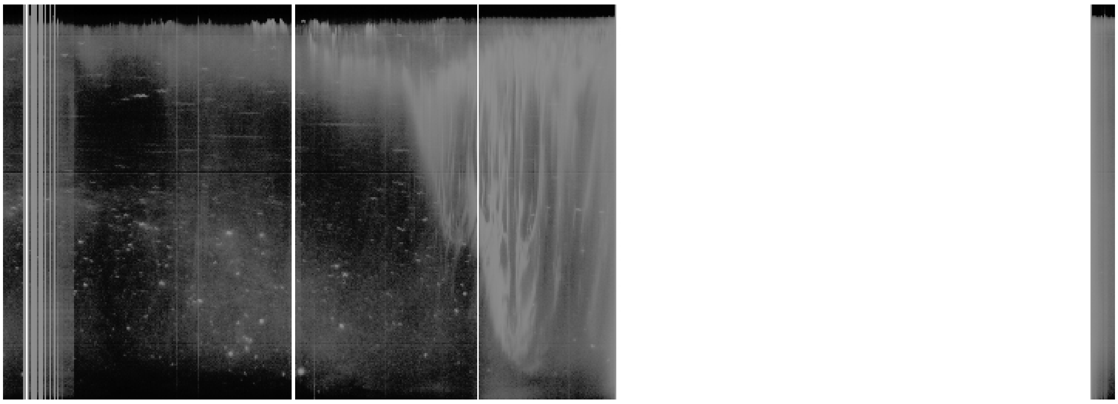

Figure 2: The keogram included in the data for January 1, 2011.

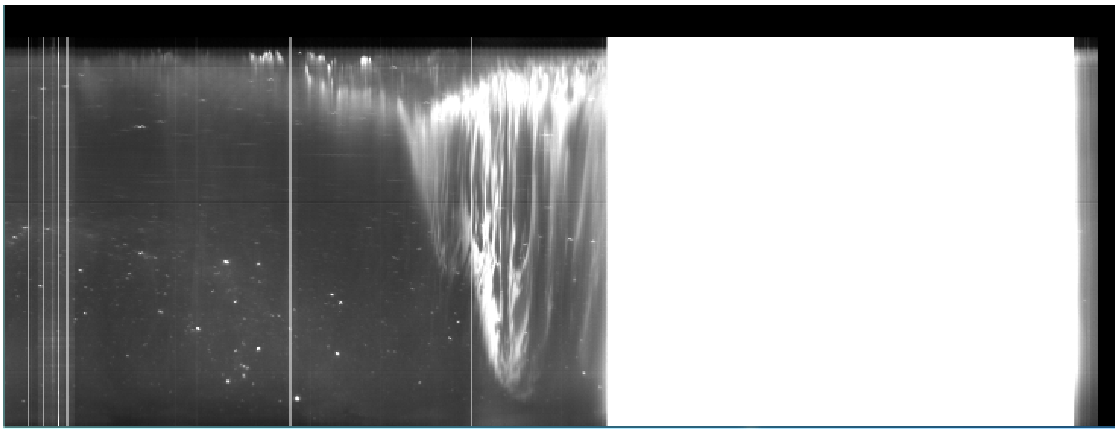

Figure 3: My attempt to create the same keogram. The scaling is different but the two images seem to represent the same information.