JWST GENERAL TRAINING (DATA CUBES)
Research & Instrument Analysis Branch

Versions:
1.0: Rachel Anderson, Misty Cracraft (2011)
1.1: Charles-Philippe Lajoie (August 23, 2012)

1. INTRODUCTION

This document briefly summarizes the format and manipulation of IR data cubes and is meant as an introduction only. In particular, the data used here is taken from the MIRI Verification Model testing that took place in 2008 and we shall focus on this instrument only. Those interested in learning more on methods for reading IR detector data should complete the JWST Data Project.

The Mid-Infrared Instrument (MIRI) on JWST is equipped with Si:As infrared detectors that cover the spectral range from ~5 to 28 microns. The size of the detectors is 1024x1032 pixels, consisting of both active and reference pixels, and it uses four amplifiers as well as reference

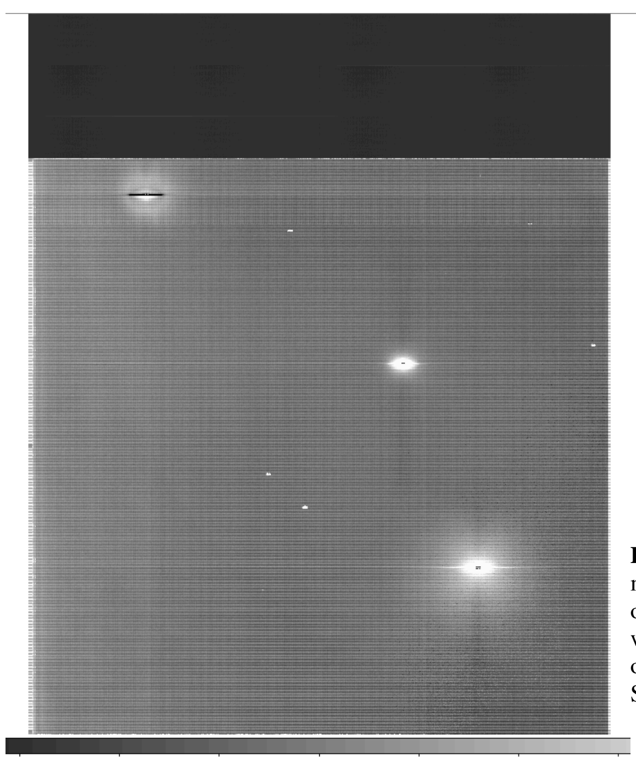

Figure 1: Typical MIRI image (last frame of miri small.fits) showing the reference outputs (top of image) as well as the 4-pixel wide reference pixel columns on both sides of the detector. Taken from Lajoie 2012 STScI RIAB Training Report.

outputs (MIRI only). Figure 1 illustrates a typical output image from MIRI, where the reference pixels are found on the first and last four columns and the reference output at the top (rows 1025 to 1280). The data you will be using for the problem sets is in this format.

Unlike CCDs, which accumulate charges and read them out only once the integration is over, the technology of IR detector allows for the charge in each pixel to be continuously read out throughout the integration (see Figure 2). Doing so creates so-called *ramps*, i.e. the increase of charge in one pixel as a function of time, and fitting the slope of a ramp yields the count rate (counts per second). It therefore follows that objects of different brightness will have different slopes and that variable phenomena (e.g. flares, cosmic rays, etc) will display non-constant slopes. Apart from readily providing the count rate from a source, using such non-destructive readings has the advantage of significantly reducing the readout noise.

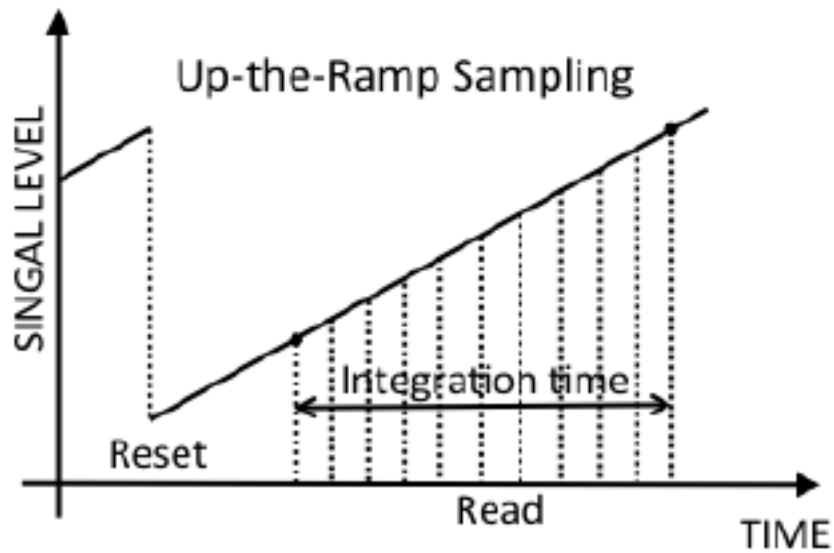

Figure 2: Technology in infrared detectors allows for the charge in each pixel to be read out throughout the integration (vertical dashed lines). The Up-the-Ramp method, also referred to as Multiaccum, is often used when reading out IR detectors. Taken from Lajoie 2012 STScI RIAB Training Report.

To summarize, after one integration, every pixel on the detector has its own ramp, just like in Figure 2. Moreover, every time we read out during the integration, we get an image just like in Figure 1. The problem sets in the next section will help you better understand this concept.

Of course, there are different ways to read IR data cubes, each with their own pros and cons. For example, you can change the frequency at which you read the pixels (Fast vs. Slow modes) or use reference pixels and outputs to further decrease the readout noise. We will cover these different methods and pixels in the JWST Data Project, but for now treat reference pixels and reference outputs the same as the active pixels.

2. EXERCISES

To ensure your understanding of IR data cubes and ability to manipulate them, complete the following exercises. Do not hesitate to discuss with each other and/or ask me questions if you get stuck. Once you are done, show me your answers so we can compare and check you got the answers right.

The data you will need for the exercises can be found in: **/user/lajoie/Training/JWST/**

MIRI_VM2T00003329_1_IM_S_2008-09-11T10h04m18.fits
MIRI_VM2T00003582_1_IM_S_2008-09-14T11h02m41.fits

Note that the pixel coordinates given below refer to IDL/Python coordinates, which are 0 indexed. To get the equivalent DS9 coordinates, which are 1 indexed, just add 1.

PROBLEM 1

For each file, do the following. In PyRAF, use *catfits* to look at the file structure. How does it differ from HST data? Look at the header of the file to check how many frames and integrations there are. What is the time between frames?

Load the file into ds9 and look through the data cube. Familiarize yourself with the concept of ramps (aka *multiaccum*) and determine the frame number(s) where the reset(s) occurs. How many integrations are there? Find the location of the point source. What is the pixel location? (*Hint*: There are a few bad pixel regions on the detector, and the point source will be much fainter than these.)

PROBLEM 2

Using the programming language of your choice, do the following:

Load MIRI_VM2T00003582_1_IM_S_2008-09-14T11h02m41.fits. Plot the values of pixels (x,y) = (2,657), (376,693), and (282,441) through the cube to see how they behave 'up-the-ramp'. Locate these pixels on the image using ds9. Why do they behave differently?

Subtract off the first frame from all subsequent frames.

For each pixel in the frame, fit a line to the 'ramp' and find the slope. Write out the final slope image. What are the median, average, and standard deviation of the slope image?