Skimmed 2D Images

Jeren Suzuki

Last Edited May 1, 2013

Contents

1 Introdution

Now that we've got the *basic* framework of finding the centers of 3 suns, let's make the code flexible enough to handle each scenario:

1	2	3
1		3
1	2	
	2	3
1		
	2	
		3

Where each row corresponds to a possibility of which regions will be in the image. this table already accounts for partial suns which we do not find centers of.

2 Flexibilizing Our Code

As of now, the code only works with an image with regions 1, 2, and 3 as whole. The first step to make the code work under any condition is to create a workflow that efficiently makes decisions based on how many regions/what those regions are. The first step is to check if there are any partial suns, and if so, which. Once we have isolated which regions are center-friendly, we pass the region id(s) into the program. For example, for our current setup, we'd pass reg1, reg2, and reg3 as parameters in our program. There is a small problem however where in the current setup, the auxilliary regions depend on the position of the primary region. If for some reason the primary region is cut off, we're going to have some problems here.

3 Dealing with < 3 suns

How do you tell the difference between 2 suns where one is $\approx 50\%$ lower in intensity than the other? It is either a pair of region 1 and 2 suns or a pair of region 2 and 3 suns. Furthermore, how does one tell the difference from a region 1 and 2 sun that is somehow oobscured to 50% brightness and a region 2 and 3 sun at normal brightness? Unless there is a way to know whether or not something is artificially making the suns dimmer, it's hard. However, if we know that and we know the approximate thresholding, we can make guesses on which region is which.