7/1/2019 SubpixelCode

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.pyplot import \*

from astropy.utils.data import get\_pkg\_data\_filename

from astropy.table import Table

from astropy.io import fits

import glob

import batman

import lmfit

import corner

# directory='/Users/annaburkholder/exp\_det\_scripts/visit23\_defringed/'

# directory='/home/ian/Desktop/WebbData/visit23\_defringed/'

directory='/visit23\_defringed/' #Change directory to proper location

number\_of\_images=43

#Load images into a list

list=glob.glob(directory+"\*.fits")

#print, first image in list

print(list[0])

####Example load first fits image

hdul=fits.open(list[1])

#Get MJD mid time of exposure from Header, which has start and end MJD times

mjd\_start=hdul[0].header['EXPSTART']

mjd\_end=hdul[0].header['EXPEND']

mjd=(mjd\_end+mjd\_start)/2.

#maybe no period?

print(mjd)

#load fits file image into an array called 'data'

data = hdul[0].data

data.shape #size of image

data.dtype.name #type of image

print(np.sum(data)) #total counts in image

#close fits after loading in data needed

hdul.close()

#load all fits images

#Arrays created for MJD time, and the white light curve total\_counts

index\_of\_images=np.arange(number\_of\_images)

mjd=np.zeros((number\_of\_images))

total\_countsPixel=np.zeros((number\_of\_images)) #Collects the photon count of the aperture, without factoring in subpixels

total\_countsSubPixel=np.zeros((number\_of\_images)) #Collects the photon count of the aperture, factoring in subpixels

for i in index\_of\_images:

img=list[i]

print(img)

hdul=fits.open(img)

mjd\_start=hdul[0].header['EXPSTART']

mjd\_end=hdul[0].header['EXPEND']

mjd\_image=(mjd\_end+mjd\_start)/2.

mjd[i]=mjd\_image

print(mjd[i])

data = hdul[0].data

print(np.sum(data)) #total counts in image

MinusSubPixel=np.sum(data[59-7,0:1024]\*0.1) #Add the subpixels along the row below the lower aperture threshold

PlusSubPixel=np.sum(data[59+7,0:1024]\*0.1) #Add the subpixels along the row above the upper aperture threshold

total\_countsSubPixel[i]=np.sum(data[59-6:59+6,0:1024])+MinusSubPixel+PlusSubPixel #total counts in 12 pix wide aperature around pixel 59 in image, including subpixels

total\_countsPixel[i]=np.sum(data[59-6:59+6,0:1024])

#Print out the total error for the full pixel and subpixel sums respectively to compare

total\_errorPixel=np.sqrt(total\_countsPixel)

print(total\_errorPixel)

total\_errorSubPixel=np.sqrt(total\_countsSubPixel)

print(total\_errorSubPixel)