Gravitationally Lensed Quasar Project

<u>Intro</u>:

In this project, we studied gravitationally lensed quasars' shapes and how to find them in an efficient way. A gravitationally lensed quasar is basically a tiny mirage in the sky (it's so small that it is equivalent to us looking at a dime from 2 miles away). Very far away, on the edge of our universe, there are many millions of large galaxies (quasars) that project light in every direction. However, between these galaxies and earth, on occasion, there is a chance that another galaxy is in the way that bends the light

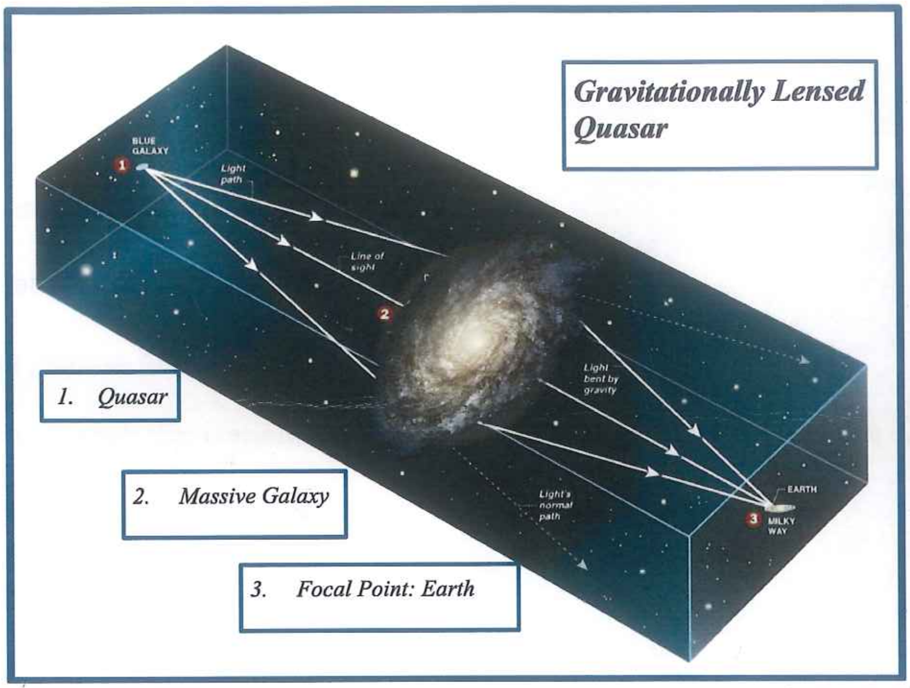
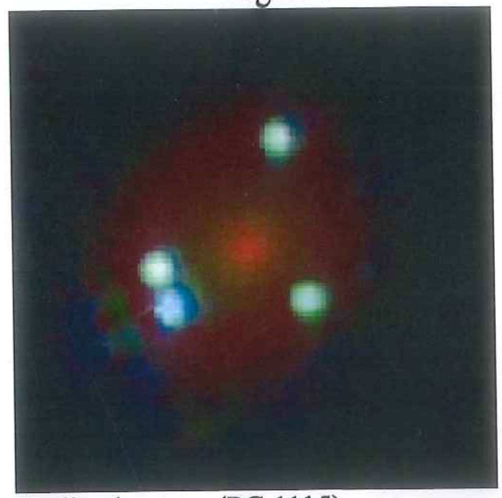


Figure 1

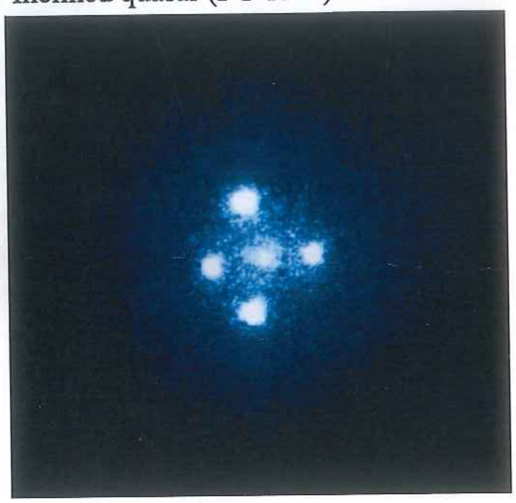
that we see because of its gravitational pull (see figure 1). Therefore, the quasar is still visible on earth, but it will create a mirage of 4 different galaxies because of the gravitational pull of the closer galaxy. (This is called a "quad grav lens"). There are other types of grav lenses where you can see 2 or 6 images as well, but the quad lenses are the most interesting.

There are 3 different patterns that occur for a quad lens quasar: cross, inclined, and cats-paw (see figure 2). The pattern we see depends on where the galaxy is positioned with respect to the background quasar. There are only about 500 gravitationally lensed quasars in visible from Earth, however only about 200 of them have been discovered. Out of the gravitationally lensed quasars, only about 20 of them are quads.

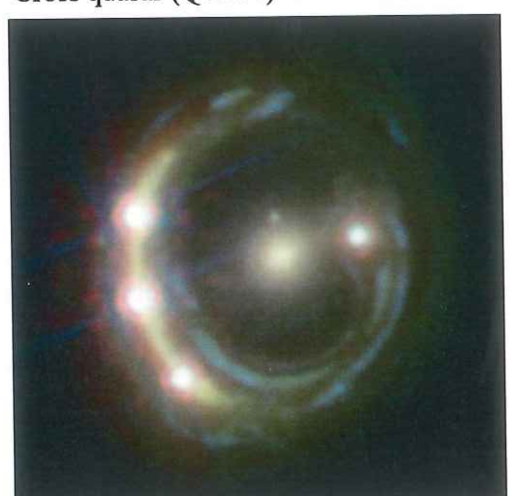
Figure 2



Inclined quasar (PG 1115)



Cross quasar (Q 2237)



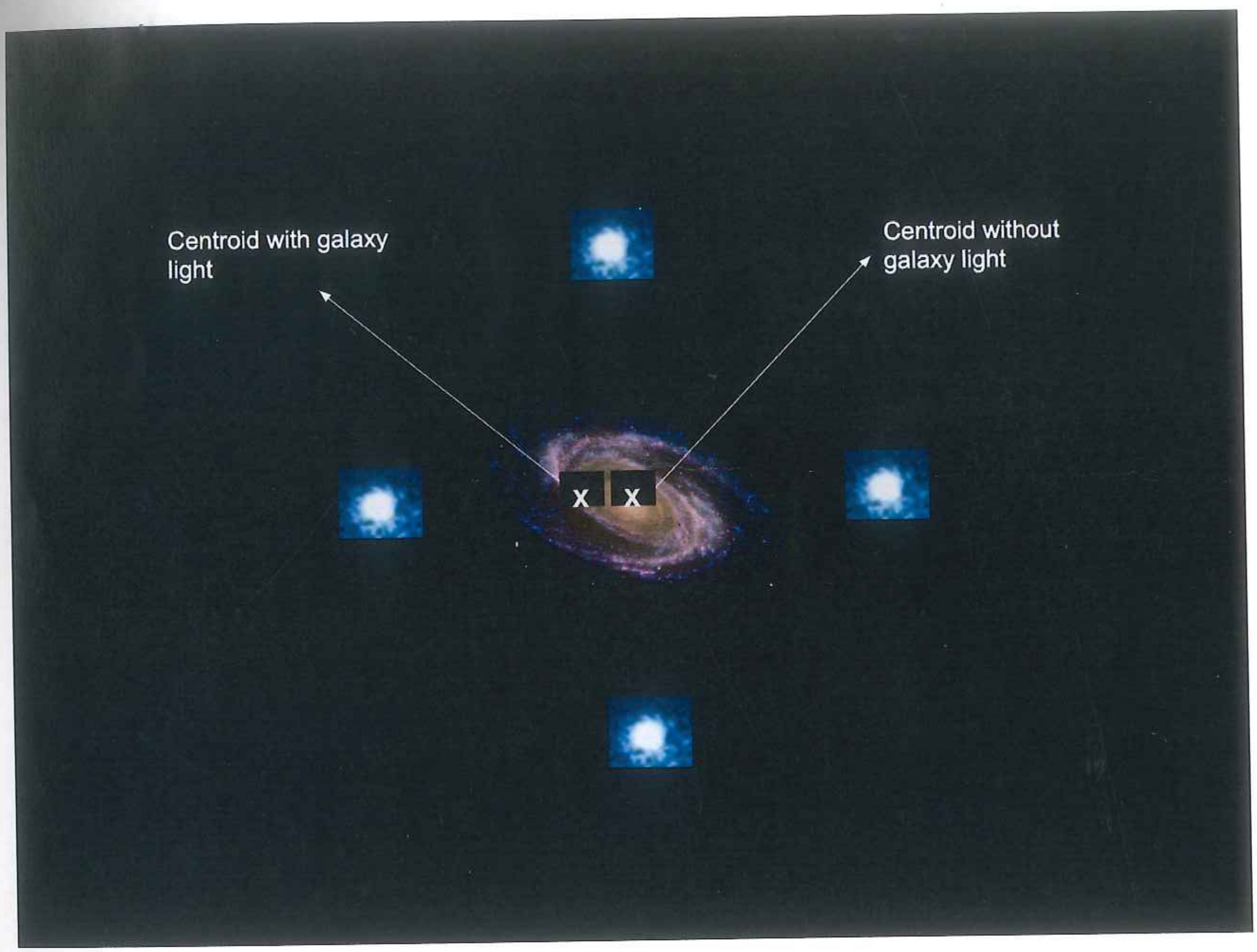
Cat's paw quasar (RXJ 1131)

Method:

Grav lenses are not so easy to find. However, there are 2 surveys that take pictures of the sky in great detail. These surveys, GAIA and Pan-STARRS, are both helpful for finding quasars. GAIA is a satellite that is in orbit around the Earth and continuously takes pictures of space. Pan-STARRS is a telescope in Hawaii that constantly takes pictures of space as well. When GAIA takes pictures of the universe, it captures all visible light at the same instant. When Pan-STARRS takes pictures, it uses filters such as color filters (blue and red for example) and infrared light filters that only admits a slice of the light in. We will use these surveys to pick out certain "blobs" in the sky that could potentially be a quad gravitationally lenses. We will then evaluate the pictures by using certain filters that GAIA and Pan-STARRS provide. These GAIA images will be helpful in first identifying a quasar. Using the color filters of the Pan-STARRS survey will also be very helpful in identifying quad grav lense quasars. The center galaxy is usually red/orange, and the quasar is blue. The blue filter will help locate the 4 locations of the quasar (and not the center galaxy), which will determine the type of quasar (cross, inclined or cat's paw). The red filter will show the center galaxy and not the quasar. However, the main reason for these filters is to determine if the "blob" in the picture is really a quasar, and not something else. With the blue filter, there is a feature that we can use to pinpoint the average center of all light of the quasars. Then, using the red filter, we can pinpoint where the galaxy is positioned. The way we can tell if it's a quasar is if the center of the quasar light and the center of the the center galaxy are not the same, because this will prove that the center galaxy is truly in the way of the distant quasar and is bending the light. (This shift of the light will be very small on a computer, but is fairly large in life-size). This helps us find inclined and cat's paw quasars because in these quasars, the quasar images aren't evenly dispersed. (See Figure 3 on page 4 for a diagram). Therefore, it is impossible for us to tell, if there is a cross shaped quasar, if it is actually a quasar because the center of the light wouldn't change anyway.

Our goal for this project was to make a code that would calculate the center of an image (for instance a quasar), based on the light photons in each pixel. Dr. Morgan first started out this project by giving me little coding challenges in order to get me used to the language of coding (python) because I never used it before. The first challenge was to write a list of 3 numbers and find the average (page 5). Later on, I was instructed to find weighted averages (similar to finding weighted test score averages) (page 6). These challenges were very helpful because practicing with this new language of code was very difficult in the beginning but as I completed the challenges, I got used to the format and commands in python. Finally, I was instructed to make a code that finds the x and y center of any image based on the brightness (page 7&8). This code can is crucial to our project because this code helps determine if the image taken from either the GAIA or PanStarrs actually shows a quad grav lens quasar because we will be able to tell if the center of light from the image is different from the center of light from the middle galaxy.

Figure 4



The centroid moves in the different filters. This is how we can tell if its a quad grav lense quasar.

```
myList=[70,80,90]#test score array


def getAverage(myList): #unweighted average function
    thesum=sum(myList)
    thelength=len(myList)
    avg=(thesum/thelength)
    return avg#asking the function to return the average


print(getAverage(myList))#print the average
```

```python
myList=[70,80,90]#test score array


myWeights=[1,1,10]#test weight array



def getAverage(myList): #unweighted average function
    thesum=sum(myList)
    thelength=len(myList)
    avg=(thesum/thelength)
    return avg#asking the function to return the average


print(getAverage(myList))#print the average



def getSum(myarray):
    sum = 0
    for i in range(0,len(myarray)):#denominator "for" loop
        sum+= myarray[i]
    return sum

def getWeightedAverage(myList,myWeights):#weighted average function
    denom=0 #defining the starting value of denom
    num=0 #defining start number of num

    for i in range(0,len(myList)):#numerator "for" loop
        #num+= (myList[i]* myWeights[i])
        num+= i * (myWeights[i])

    wAverage = num/getSum(myWeights)
    return wAverage#asking the function to return the weighted average
print(getWeightedAverage(myList,myWeights))#print weighted average
```

See page 8 for what the "myImage7" would look like on a gridded image.

```
myImage7 = [ [2,3,4,3,2], [3,4,5,4,3], [2,3,4,3,2], [1,2,3,2,1], [0,0,0,0,0] ]


def getIndexAverage(myImage):
    xnum=0
    xdenom=0
    ynum=0
    ydenom=0
    for h in range(0,len(myImage)):
        for j in range(0,len(myImage[h])):
            xnum+= (j*(myImage[h][j]))
        xdenom+=sum(myImage[h])
    xcenter=xnum/xdenom

    for g in range(0,len(myImage[0])):
        for f in range(0,len(myImage)):
            ynum+= (f*(myImage[f][g]))
            ydenom+= myImage[f][g]
    ycenter=ynum/ydenom



    return [xcenter, ycenter]

print(getIndexAverage(myImage7))
```

Center of this image is at *[2.0, 1.32142857142857I4]* (X,Y)

Y-Possition \ X-Position	0	1	2	3	4
0	2	3	4	3	2
1	3	4	5	4	3
2	2	3	4	3	2
3	1	2	3	2	1
4	0	0	0	0	0