A PCA-based Strong Gravitational Lens Finder

VIVEK SANGWAN 12D070033

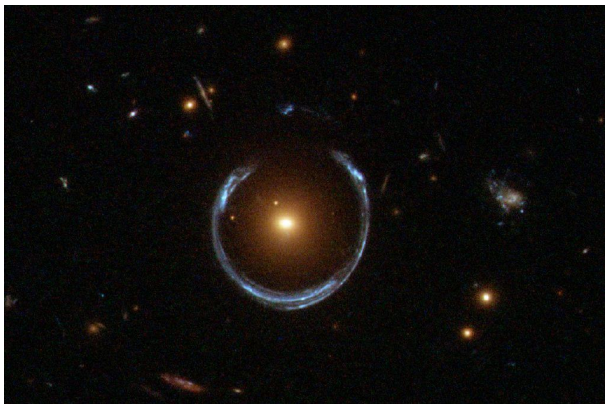
KUSH MOTWANI 12D070018

KOTWAL ALANKAR SHASHIKANT 12D070010

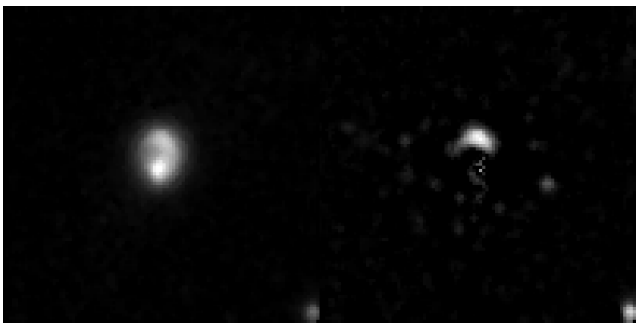
November 3, 2014

1 Introduction

Gravitational lensing is one of the most visible implications of Einsteins General Theory of Relativity. This effect predicts that light is bent by gravitational fields. In particular, if we have a heavy object in directly front of a source of light in our line of sight, this effect causes ring- and arc-like features like those shown below:



Our job here is to identify these features without modifying their size and shape. The size and shape of a ring contains vital information about the configuration of the system, especially the mass of the lens. If our method is to be useful for further processing, we must recover the arc as it was in the original image. We exploit PCA and the fact that lenses are rare objects in order to efficiently reconstruct the central galaxy, subtract it from the image and get the arc. A sample output, for instance, would be:



The left image is the original, and on the right the central source has been subtracted to leave only the ring.

2 Method

The method we follow is a slightly modified version of the paper *Joseph, R., Courbin F. et al. 2014, A PCA-based Automated Finder for Galaxy-Scale Strong Lenses*. You can find the paper here.

The fact that the lensing patterns are rare among galaxies in general is used. We construct a PCA basis for aligned galaxy images using the train dataset. In this basis, the galaxies in the frame are represented adequately, and the lensing features, being rare, are under-represented. As a consequence, reconstructing the galaxy image from this basis yields mostly only the bright central core of the object. Subtracting the reconstructed image from the original image is therefore expected to yield the (non-reconstructed) lensing features. Denoising, thresholding and segmentation of the subtracted image can give an image (andhopefully measurements) of the features that further help in lens mass estimation.

3 Implementation

We implemented the algorithm in Matlab.

3.1 Image Acquisition and Denoising