Final Year Project Proposal

Machine Learning analysis of Gravitational Lensing Phenomena

Michał Staniaszek
Supervisor: Dr Peter Tiňo

May 7, 2012

Description

Gravitational lensing is one of the many corollaries of Einstein's general theory of relativity, and he is often associated with it. Indeed, Einstein quantified the phenomenon in 1936, but its first mention is by Russian physicist Orest Khvolson in a 1924 paper. Confirmation of its existence came in 1979 when the gravitationally lensed quasar SBS 0957+561 was accidentally discovered by British astronomer Dennis Walsh his colleagues. Since then, many lensed objects have been discovered, but the field is still relatively young.

This phenomenon causes multiple images of the same object to be visible at the same time. The projection of these multiple images is caused by the bending of light due to a massive object, such as a cluster of galaxies or a black hole in between the object and the observer. This massive object is referred to as the gravitational lens. According to general relativity, any object with mass warps space-time. Since light propagates through space-time, the light rays travel across the warped space, and are bent. As a consequence, rays of light coming from various different directions can reach the same point in space, leading to the object appearing multiple times, rather than just once, as is normal for an object whose light is not bent.

Each image seen by the observer is in fact the object at a different time. It is common knowledge that the light from astronomical objects is not, for want of a better word, 'new'. The light must travel millions, often billions of light years to reach the observer. With an unlensed light ray, this is not so much of a problem—it just means that we see the object how it looked as many years ago as it took the light to reach our measuring instrument. However, when light is gravitationally lensed, the light from each image has a different travel time to the others. In the case of SBS 0957+561, where there are two images, the delay between one image and the other has been estimated to be 417±3 days.

Most gravitationally lensed objects have time delays on the order of hundreds of days. However, there are also some objects which due to their relatively close proximity to earth, or the way that they are lensed, have delays on the order of hours. In this project, we will be looking at objects like these.

Telescopes use various techniques to capture a certain range of the electromagnetic spectrum. Invented in the Netherlands in the 17th century, the optical telescope was the first type of telescope. Initially using glass lenses, the design was quickly improved and replaced these lenses with mirrors, leading to the birth of the reflecting telescope, variations of which are still used now, albeit with great improvements in the technology. In the 20th century, telescopes capturing the infrared, radio and other bands of the electromagnetic spectrum were invented. Although gravitational lensing can also be observed in these bands, this project will focus on the visible part of the spectrum. Reflecting telescopes gather photons propagating from the objects in the area being observed. Depending on the brightness and the distance of the object from the telescope, it is usual to receive a few photons per millisecond. Most telescopes use track how many photons arrive in the space of a few seconds, with a so-called 'bin' for each of these intervals. The count of the photons that arrived in a single time period is called the flux. It is also possible to get two pieces of data on each individual photon—its energy, or wavelength, and its arrival time. This will be the raw data available to us in this project.

Approach

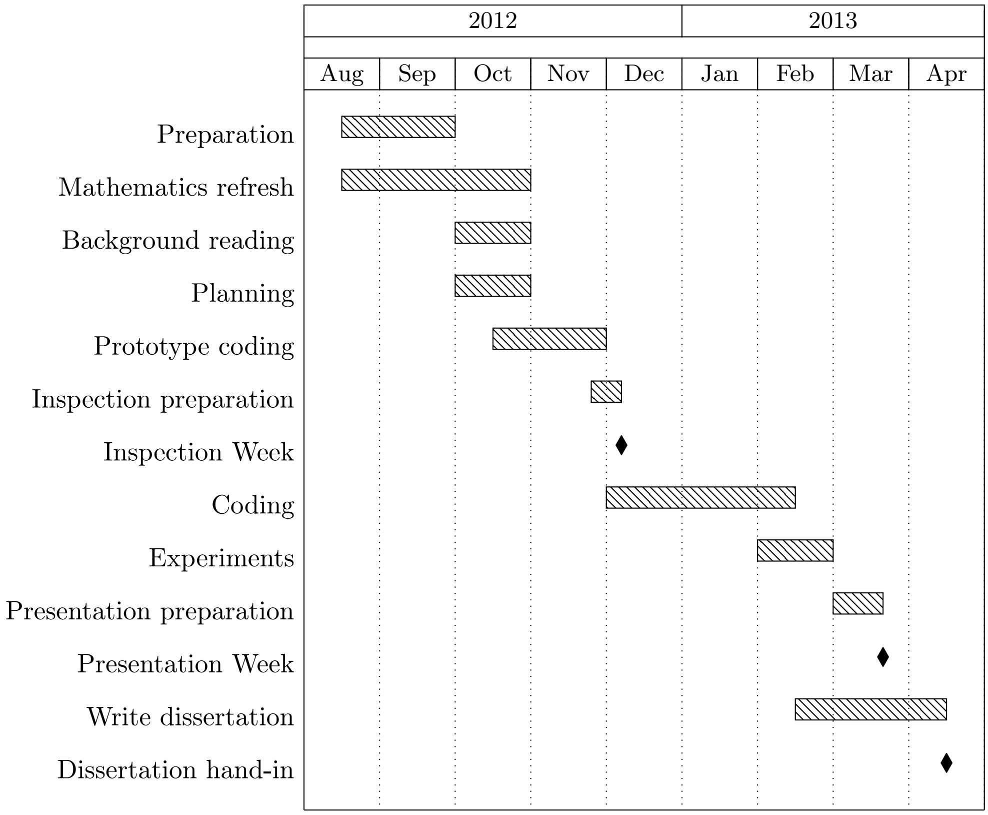
The final goal of the project will be to take two streams of photons and determine whether they are coming from the same gravitationally lensed object. The ability to determine this is dependant on the variability of the source. When working with delays on the order of hours, it is necessary for the source to be variable on the order of thirty minutes or so in order to get good results. It may be the case that it is possible to get good results when the source does not vary as much, but this remains to be seen, and could become an additional task should the final goal be completed ahead of time. Each of the two streams of photons is generated by a different non-homogeneous poisson process. This leads to each stream of photons being represented by a poisson distribution. Another goal would be to find, given these two distributions, whether they could have been generated by the same poisson process.

The approach will be based around non-homogenous poisson processes to model the photon streams, and the analysis of the data will be done using kernel methods. There are several stages of the project at which it is possible to say that something useful has been achieved:

- Generation of artificial photon data streams through poisson processes
- Fitting a model to a single stream of photon data
- Fitting a model to coupled streams of photon data
- Performing various experiments on simulated coupled streams
- Performing experiments on real world data

It is intended that there be a meeting each week between the supervisor and student in order to review the work done in the past week, and to discuss any difficulties encountered, as well as the discussion of a plan for the next week's work. Notes will be taken each week by the student, describing the various approaches to an encountered problem, and the reason why a specific approach was chosen. In addition, the progress made each week will be briefly described. This will lead to the dissertation being easier to write, as each decision and the reason for it will have been written down.

Timetable



Preparation Learning C and C++, as well as meetings with supervisor to finalise various project details.

Mathematics refresh Revision of any mathematical techniques necessary for the project, learning of techniques that have not been previously learnt.

Background reading Reading of material relevant to the project. This will involve material from both computer science and physics.

Planning Discussion with supervisor to decide the initial approach to the project. When this has been decided, more specific discussion and decisions on the code structure and the like.

Prototype coding Implementation of the code structures from the planning stage to see if they are viable.

Inspection preparation Look at the current state of the project, and decide whether it is proceeding relatively satisfactorily. If not, identify reasons for this. Note down what has been achieved, and the direction of the project from this point forwards.

Coding Completion of the code base for the project. Some testing will probably also be necessary.

Experiments Carrying out of experiments using simulated and real world data.

Presentation preparation Preparation of presentation slides and data to be used in the demonstration. Practise the presentation multiple times alone to make sure of the flow.

Write dissertation In the initial stages, thinking about what to write in each section and how to present the data will be the main aim. Writing several bullet-points outlining the main points for each section would likely be a useful guide. The dissertation will be written using LaTeX, and so preparation of some sort of template may be necessary. The later stages will involve compiling data from experiments into elucidative diagrams and/or graphs, and the fleshing out of the basic explanations in the first draft. There should also be some time for proofreading by various people in order to ensure that there are as few mistakes as possible and that there is a good flow.

Hardware & software

It is intended that the C programming language be used as the main language for the project, but this is yet to be decided. The main reason for changing the language would be if it was decided that the project would benefit from the use of object orientation, since C does not support this. Other languages in consideration include Haskell, C++ and Python. If the need arises, Matlab may also be used, but it would be preferential to complete the project without its use. There will likely be no hardware requirements for this project other than a system which can run the finished code.