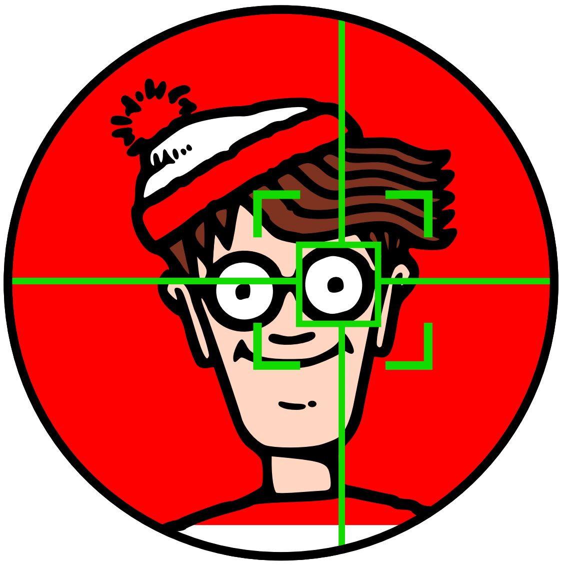
Project Preparation Report
Where's Wally?

Cian Booth

March 7, 2013



Contents

1 Introduction

This is an introduction Page

2 Previous Dissertation Reviews

When reading previous dissertations, I tried to read a range of papers, to maximise their benefit to me. Firstly, I chose to read a paper that was strongly related to my own, *'High Performance Computer Vision'* by Benjamin Eagan. This paper has the same topic, Computer Vision, as my own. As such, I could learn a good amount about any hints or difficulties found with Computer Vision in the high performance domain. My next choice was *'Parallelization and Optimization of a Tax and Benefits Model'* by Thomas McClintock. This has almost no relation to Where's Wally, being a simulation of how taxes affect various groups of people. Thus, I could learn from this the general things that I should note in order to complete this project efficiently. However, my previous two papers were both on the list of dissertations which achieved a distinction. I therefore chose to read another paper, *'Novel energy efficient compute architectures on the path to Exascale'* by Ioan Corneliu Hadade. From this paper, I could compare the differences between a paper that achieves a distinction, and one that achieves something between that and a pass. Hopefully I could extract that which makes the difference, and then attempt to do the same.

2.1 High Performance Computer Vision

Eagan's paper mostly deals with two topics; object tracking in real time videos, and disparity mapping. The first is useful for having more natural human-computer interactions. The second is for allowing computers to map 3D areas from 2D images. These are not the areas of computer vision that I will be concerning myself with, but are still in the general domain.

The good qualities of the paper were apparent, presumably leading to the distinction grade it received. The quality of English throughout the paper was entirely understandable. The figures included were engaging, and generally it was easy to grasp what they were showing. The nature of the project that Eagan had chosen also strongly highlighted the importance of Risk Assessment. When choosing an algorithm to use, Eagan was very thorough in explaining why it was the optimum choice.

However, I felt that the paper also had a few failings. The introduction section felt quite long, and took quite a while to get around to basic message of the section. This message was roughly "Supercomputers are not used for Computer Vision because they are too expensive to run, but cheaper methods would be". I also felt that the explanation of the project goals was obfuscated within the lengthy introduction. Many figures were of processed images, but there were no figures of the original, making it hard to see the true effectiveness of the code. Some figures also showed errors in the processing. It would be useful if these captions briefly explained what caused these errors. Finally, some of the lines in the graphs included were very hard to distinguish from each other. The makes the plots somewhat indecipherable.

The paper also contained other things that I had not considered myself. These are neither good, nor bad, but their impact should be considered.

- The paper it written in something very close to default LaTeX.
- Within the introduction, were standard definitions of speed-up and efficiency.
- Includes a list of figures and tables
- Each section is concluded with a brief summary of what came before
- Eagan chose to use the HSV colour scheme over the RGB scheme.
- Benchmarking in computer vision is not standardised yet

2.2 Parallelization and Optimization of a Tax and Benefits Model

2.3 Novel Energy Efficient Compute Architectures on the Path to Exascale

2.4 Conclusions Drawn

3 Work Plan

As with most projects, there is a set time that I must complete the code and report within. According to the MSc. Programme Handbook[1], the start of the dissertation period is on the 3rd of June 2013. The dissertation must be submitted by the 23rd of August 2013. This gives me 88 days to produce my project.

My project, as presumably do many other *'from scratch'* HPC projects, has two main regions areas of development: the linear code and the parallel code. As such it makes sense to divide my time into two regions, linear and parallel. This division is somewhat coarse, however, so each region will be divided into three new sections.

The first will be *Design*, wherein I will be deciding the overall format of the code. This will include, what programming language, libraries, functions and the actual method of solving Where's Wally. The next section will be *Creating Tests*, where I will design the tests to ensure functionality of the solution. The final section will be *Implementing the Solver*. This will be the actual production of code that satisfies the tests I have already created.

If there is any spare time at the end of the project, a fourth section will be used; *Optimisation and Outreach*. This will be simply improving the speed of the solution I have developed. If the program can solve Where's Wally problems at a near human level, then I will work on improving the output of the program. The end goal at this point, would be to produce something that could be used for HPC Outreach.

3.1 Breakdown

The breakdown of my work plan follows. Each section has built in time for weekends, plus some stretch for running overtime.

Linear (~30 days)

Design (4 days) As I am familiar with the problems of a Where's Wally solver, it is hoped that 4 days would be enough to design a solution.

Create Tests (6 days) My problem lends itself to a relatively simple program, so testing shouldn't take more than a week.

Implement Solver (14 days) A basic Where's Wally solver should be creatable within 2 weeks.

Parallel (~40 days)

Design (4 days) Although a program utilising parallel methods is more complicated than a linear one, I hope to expand on the linear solution. Thus I have assigned the same amount of time as the linear design section.

Create Tests (6 days) Again, I hope to reuse most of the tests from the linear section, so creating tests for the parallel version should take the same amount of time.

Implement Solver (21 days)

The remaining 12 days will be entirely devoted to writing the dissertation report and presentation, however, it is intended that I will be updating these as I progress. This breakdown is visualised in Figure 1, a Gantt chart.

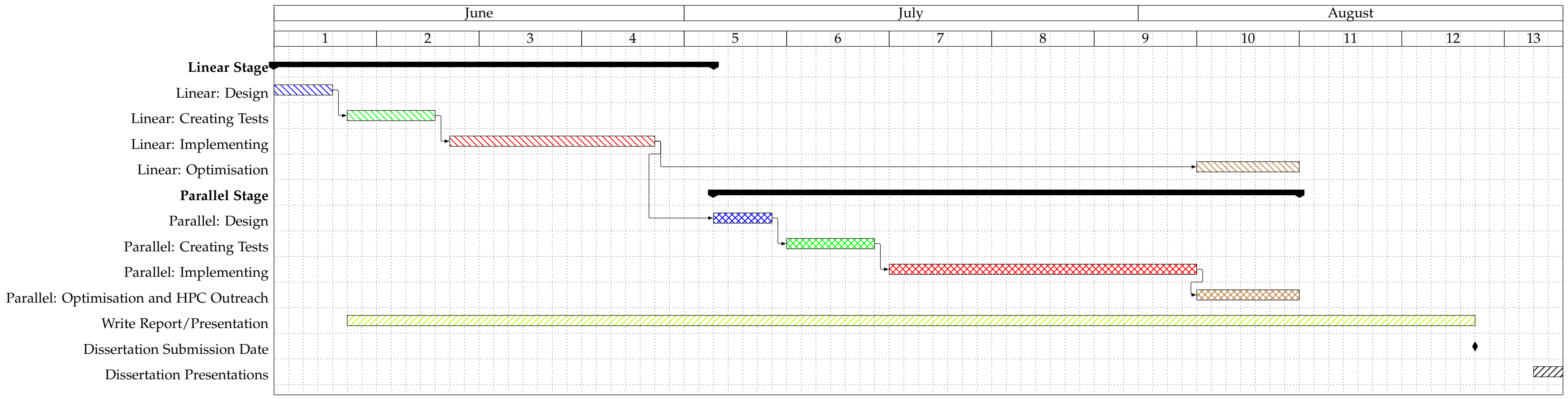


Figure 1: A Gantt Chart of how time will be divided between areas of work

4 Risk Analysis

References

[1] School of Physics and Astronomy. Msc and diploma in high performance computing student handbook, 2012. `https://www.wiki.ed.ac.uk/download/attachments/136519733/Programme-Handbook-2012+-+13+-+Final.pdf?version=1`.