Image Processing using CUDA

**October 2023**

**By**

**Dawid Kisielewski**

**Student number 202106560**

**Word count: XXXX**

Contents

[1. Project background and purpose 3](#_Toc144892521)

[1.1. Introduction 3](#_Toc144892522)

[1.2. Objectives 3](#_Toc144892523)

[1.3. Scope 3](#_Toc144892524)

[1.4. Deliverables 3](#_Toc144892525)

[1.5. Constraints 3](#_Toc144892526)

[2. Project rationale and operation 5](#_Toc144892528)

[2.1. Project benefits 5](#_Toc144892529)

[2.2. Project operation 5](#_Toc144892530)

[2.3. Options 5](#_Toc144892531)

[2.4. Risk analysis 5](#_Toc144892532)

[2.5. Resources required 5](#_Toc144892533)

[3. Project methodology and outcomes 6](#_Toc144892534)

[3.1. Initial project plan 6](#_Toc144892535)

[3.1.1. Tasks and milestones 6](#_Toc144892536)

[3.1.2. Schedule Gantt chart 6](#_Toc144892537)

[3.2. Project control 6](#_Toc144892538)

[3.3. Project evaluation 6](#_Toc144892539)

[4. References 7](#_Toc144892540)

[5. Appendix a 8](#_Toc144892541)

# Project background and purpose

## Introduction

Image processing is the process of computing an image to enhance it and or to modify the image in a way to ease the process extracting data from it. This process can often involve sharpening the image, removing graininess and or noise from an image, segmenting an image, detection of edges, and many other different processes [1]. Graphical Processing Units (GPU) have been mainly used as 3D graphics accelerators for rendering pipelines such as OpenGL or DirectX but after gaining cores people realised that moving simple parallel tasks off the CPU and onto the GPU will me more efficient. As shown by Castaño-Díez’s study in 2008 this decreases the time for image processing loads to be completed [2].

CUDA is a closed source API that allows you to write code that will run on the GPU. It was developed by Nvidia for Nvidia GPUs. It can be interfaced using many different programming languages such as C++, Python, Fortran and others.

## Objectives

To implement 3 image processing algorithms on the CPU and the GPU. Each algorithm will be from a different category of algorithms. The three categories I have picked are edge detection, noise suppression and picture segmentation. After writing both the CPU and the GPU version of each algorithm, I will proceed to test the performance of both and compare the results. All the data gathered will be compiled into one document discussing my findings and comparing the CPU performance to the performance of the GPU.

## Scope

This project will not deal with loading and saving images. For this purpose, I will be using a dedicated library that already has these functionalities built in.

## Deliverables

The culmination of this project is a document containing my findings as well as an explanation of the code I wrote.

## Constraints

For this project I’ll be using CUDA. There are many other APIs that handle GPU programming such as OpenCL.

# Project rationale and operation

## Project benefits

The project will highlight the differences between the performance of image processing algorithms running on the CPU compared to the GPU. The project would also develop my skills to write and optimize parallel code.

## Project operation

For this project I’ll be implementing the Agile workflow to help adapt to any issues I come across. I’ll be running sprints with specific targets to meet. At the end of each sprint ill reflect on the work I have done and then evaluate my Gantt chat as well as if there is any additional decomposition is required.

## Options

CUDA can interface with a variety of different programming languages. C++, Python, C#, Fortran, and others all can’t be used when doing CUDA programming. Deciding which language to use is important as it will affect the performance of the algorithms. After picking a language I’ll need to pick a library that fits the project requirements. There are many different libraries that fit such as OpenCV for C++ and Python, ImageProcessor for C#, and many others. Finally, there are many different image processing algorithms that do different things. I will need to decide which ones to implement, my target is to implement at least 3 different algorithms.

## Risk analysis

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Hazard | Risk | Mitigation | Likelihood | Severity | Impact | Residual Impact |
| Illness | Harder to focus and Concentrate on the project meaning more time wasted extending the project's time to complete | Plan time to work on tasks a bit longer to compensate for any wasted time.  Take medicine and recover quickly | High - 4 | Low - 2 | 8 | 8 |
| Data Loss | Project files being lost due to failure of hardware and or data corruption | The use of git and GitHub for cloud backups. Committing work after every task. | Low - 2 | Very High - 5 | 10 | 4 |
| Loss of Internet Service | Losing the internet will slow down my progress in investigating literature and researching any problems I encounter in my project | Going to work on campus as the internet on campus is free to use for students.  Report the issue to my landlord to get the problem fixed as quickly as possible | Low - 2 | Medium - 3 | 6 | 4 |

## Resources required

A Nvidia GPU is required to do CUDA programming, this means that I will need to develop my program and demonstrate it on a PC with a CUDA-capable GPU.

# Project methodology and outcomes

## Initial project plan

## Tasks and milestones

* Literature Review
* Write introduction to project document
* Edge Detection Algorithm
  + Research the Specific algorithm and specification
  + Implement Algorithm and test functionality
    - CPU
    - GPU
  + Analise performance and write up finders
* Segmentation Algorithm
* Noise Suppression Algorithm
* Conclusion
  + Compare GPU performance between algorithms
  + Compare CPU performance between algorithms
  + Discuss differences between performance between image resolutions

## Schedule Gantt chart

Present a Gantt chart showing a schedule for all tasks, milestones and deliverables. Show dependencies amongst tasks. If you are intending to use SCRUM or other agile methods, be sure to go to the lectures involving project planning. Your time plan should cover the entire period of your project (and will therefore include the PDD preparation as a task and the PDD itself as a deliverable).

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).

## Project control

As the project is run using AGILE, at the end of each sprint I’ll make a small reflection on what I should focus on for the next sprint. As well as this I’ll have the ability to make quick changes during each sprint.

## Project evaluation

Using a dedicated image processing library or program can be used to check the functionality of the algorithms I implemented. For the data collected, I’ll cross reference it to results of studies similar to mine.

# References

[1] Niblack, W. 1986, *An introduction to digital image processing,* Prentice-Hall International, Englewood Cliffs, N.J.

[2] Castaño-Díez, D., Moser, D., Schoenegger, A., Pruggnaller, S. & Frangakis, A.S. 2008, "Performance evaluation of image processing algorithms on the GPU", *Journal of structural biology,* vol. 164, no. 1, pp. 153-160.

# Appendix a

You may use one or more appendices to add useful reference information which may be relevant to other sections of the report. Do not use appendices simply as a way of writing more than will fit into the main document word count. If you don't need any appendices, then delete this whole section

Delete the red paragraphs and replace this one with your content (use the “Normal” paragraph style).