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)

[1.6. Assumptions 3](#_Toc144892527)

[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 a large part of computing as its used in many sectors for various applications such as medical imagery, computer vision and easing the process of gathering data from an image. Tho image processing can be done on the CPU, the parallelised nature of the GPU tends to expedite the process.

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

This project will recreate commonly used image processing algorithms on both the CPU and GPU to compare both the performance difference as well as the differences between the final output of the algorithms. All the data collected as well as the algorithms themselves will be put into a document explaining how the algorithms work.

## Scope

This project will not deal with loading and saving images. For this purpose, I will using a dedicated library what already has that functionality 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

CUDA requires a CUDA-compatible GPU. For this purpose, I will be using my personal laptop which contains a CUDA-compatible GPU.

# 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.

## 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 4 different algorithms.

## Risk analysis

What risks might affect the outcome of your project or its stakeholders? How severe are they, and what steps will you take to mitigate against them?

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

## Resources required

A Nvidia GPU is required to do CUDA programming, this means that I will need to use a PC with one.

# Project methodology and outcomes

## Initial project plan

## Tasks and milestones

Present a realistic task list for the entire project, broken down to a suitable level of detail. Indicate milestones against which progress can be monitored. Make sure you include all the deliverables you mentioned earlier.

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

## 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

How will you manage the project day-to-day? How will its performance be monitored? How will you judge if it has been successful?

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

## Project evaluation

How will you evaluate the project’s artefacts and overall outcomes? What user evaluation will you do? Do not underestimate the importance of this, and include clear details of how you will do the evaluation.

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

# References

List any sources you have used for your background and introduction here. Make sure you use the proper referencing format.

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

# 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).