Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра автоматизации обработки информации (АОИ)

Вариант 3

Отчёт к Лабораторной работе №2

по дисциплине «Параллельные вычисления и системы»

Выполнил:

Студент группы 422–M1

\_\_\_\_Белоус Г.В.

Принял:

К.т.н., Доцент кафедры АОИ

\_\_\_\_Аксёнов С.В.

Оглавление

[1 Введение 3](#_Toc116226139)

[1.1 Цель лабораторной работы 3](#_Toc116226140)

[1.2 Задание на лабораторную работу 3](#_Toc116226141)

[2 Ход выполнения работы 4](#_Toc116226142)

[3 Вывод 12](#_Toc116226143)

# Введение

## 1.1 Цель лабораторной работы

Изучить инструменты многопоточности на GPU с использованием CUDA и написание программы для обработки изображений.

## **1.2 Задание на лабораторную работу**

Написать программу на языке С/ или Python с использованием API CUDA или OpenCL. Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 1024 x 768, 1280 x 960, 2048 x 1536. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы. Сделать выводы об эффективности распараллеливания вычисления с помощью многопоточных вычислений на GPU. Оформить отчет.

Содержание отчета.

1. Титульный лист
2. Задача по полученному варианту.

(если последняя цифра 2 или 7: вариант 3)

1. Листинг программы
2. Примеры входных и выходных файлов
3. Описание результатов тестирования
4. Выводы

Загрузить цветное изображение.

Выполнить инвертирование цветов пикселей (новое значение цветового канала соответствует значению 255 – старое значение цветового канала). Выполнить свертку с фильтром, увеличивающим контраст, с каждым из цветовых каналов изображения. Ядро преобразования можно найти по ссылке: [https://docs.gimp.org/2.8/ru/plug-in-convmatrix.html]. Сохранить результат в файл.

# Ход выполнения работы

Программа для обработки изображения была написана на Python с использованием платформа Google Colaboratory для блокнотов Jupyter. Кроме среды запуска блокнотов Python и R Colab позволяет совместно использовать свободный доступ к ограниченному количеству GPU и TPU.

При запуске программы изображение загружается в форму по указанному пути с использованием библиотеки PIL, используемой для работы с изображениями, определенными данными пикселями.

Для работы с изображением был описан оператор SourceModule, который является реальной директивой для компилятора nVidia (компилятор nvcc), создающий некий модуль, который в данном случае состоит всего лишь из функции invertion\_image для инвертирования каждого спектра пикселя и sharpen\_image\_kernel для увеличения контрастности изображения с помощью фильтра “Матрица конволюции”.

Листинг программы можно увидеть в прикрепленном файле “PC\_Lab2.ipynb”.

Для проверки работы разработанной программы необходимо было взять три изображения с разрешениями a) 1024 x 768, b) 1280 x 960 и c) 2048 x 15362. Все изображения были загружены из интернета по запросу необходимого разрешения. Первое изображение, его инвертированная копия и копия с повышенным контрастом показаны на рисунках 2.1-2.3:

Изображение выглядит как птица, цветной, попугай, внешний

Автоматически созданное описание

Рисунок 2.1 – Исходное изображения разрешением с a)



Рисунок 2.2 – Инвертированное изображение с разрешением а)



Рисунок 2.3 – Инвертированное и с увеличенном контрасте изображение разрешением а)

Второе изображение, его инвертированная копия и копия с повышенным контрастом показаны на рисунках 2.4-2.6:



Рисунок 2.4 – Исходное изображения разрешением с b)

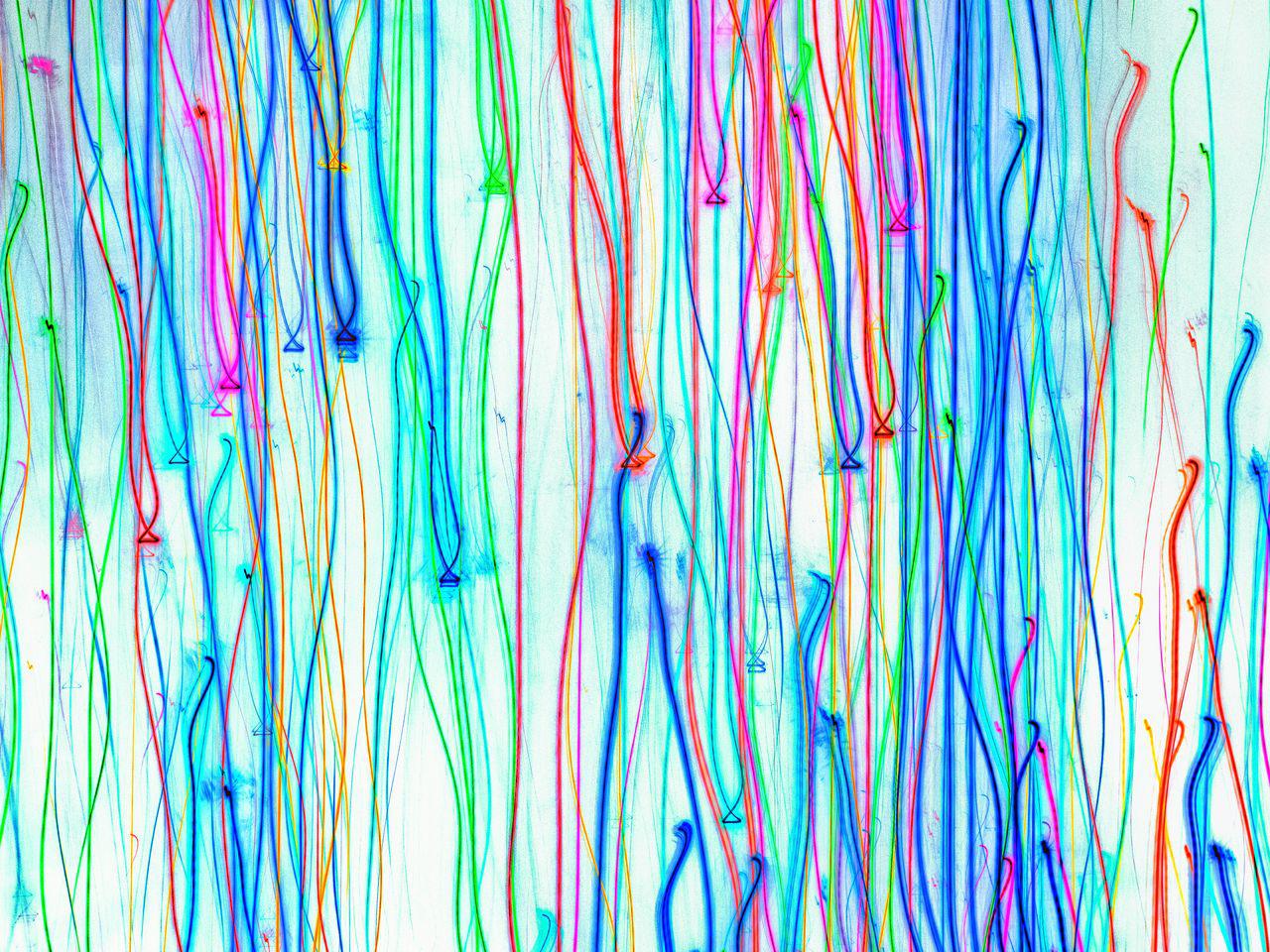


Рисунок 2.5 – Инвертированное изображение с разрешением b)

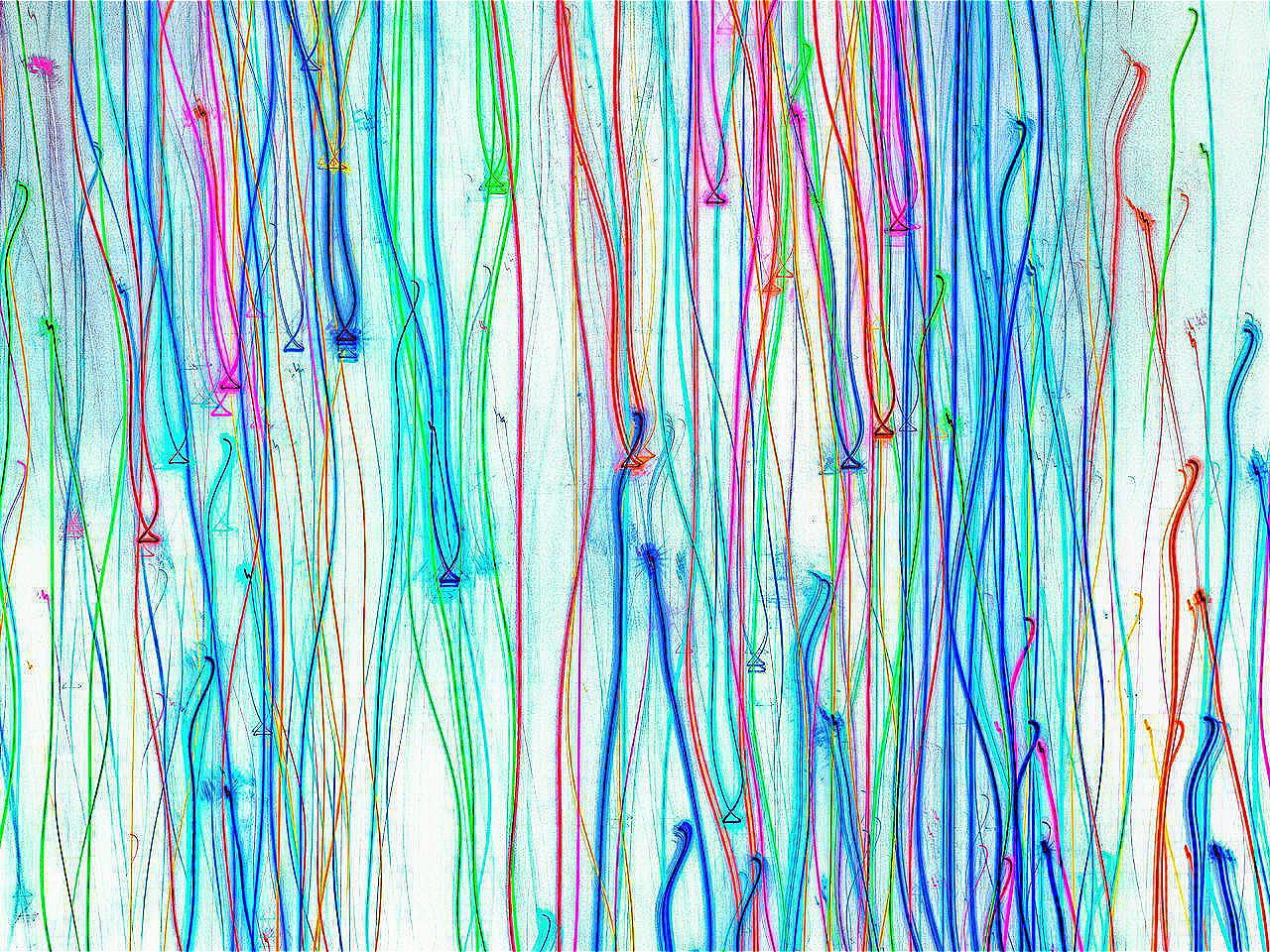


Рисунок 2.6 – Инвертированное и с увеличенном контрасте изображение разрешением b)

Третье изображение, его инвертированная копия и копия с повышенным контрастом показаны на рисунках 2.7-2.9:



Рисунок 2.7 – Исходное изображения разрешением с с)

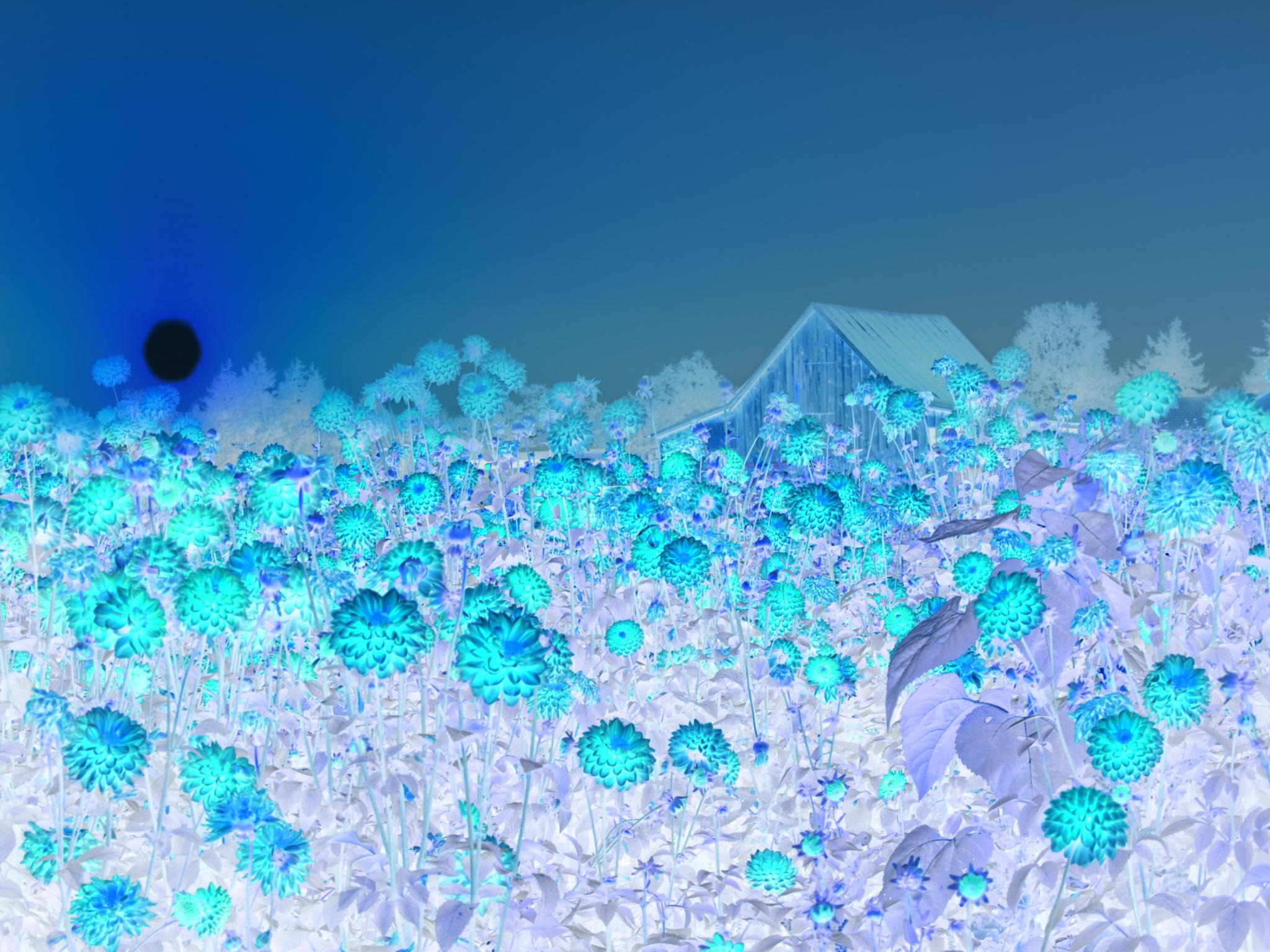


Рисунок 2.8 – Инвертированное изображение с разрешением с)

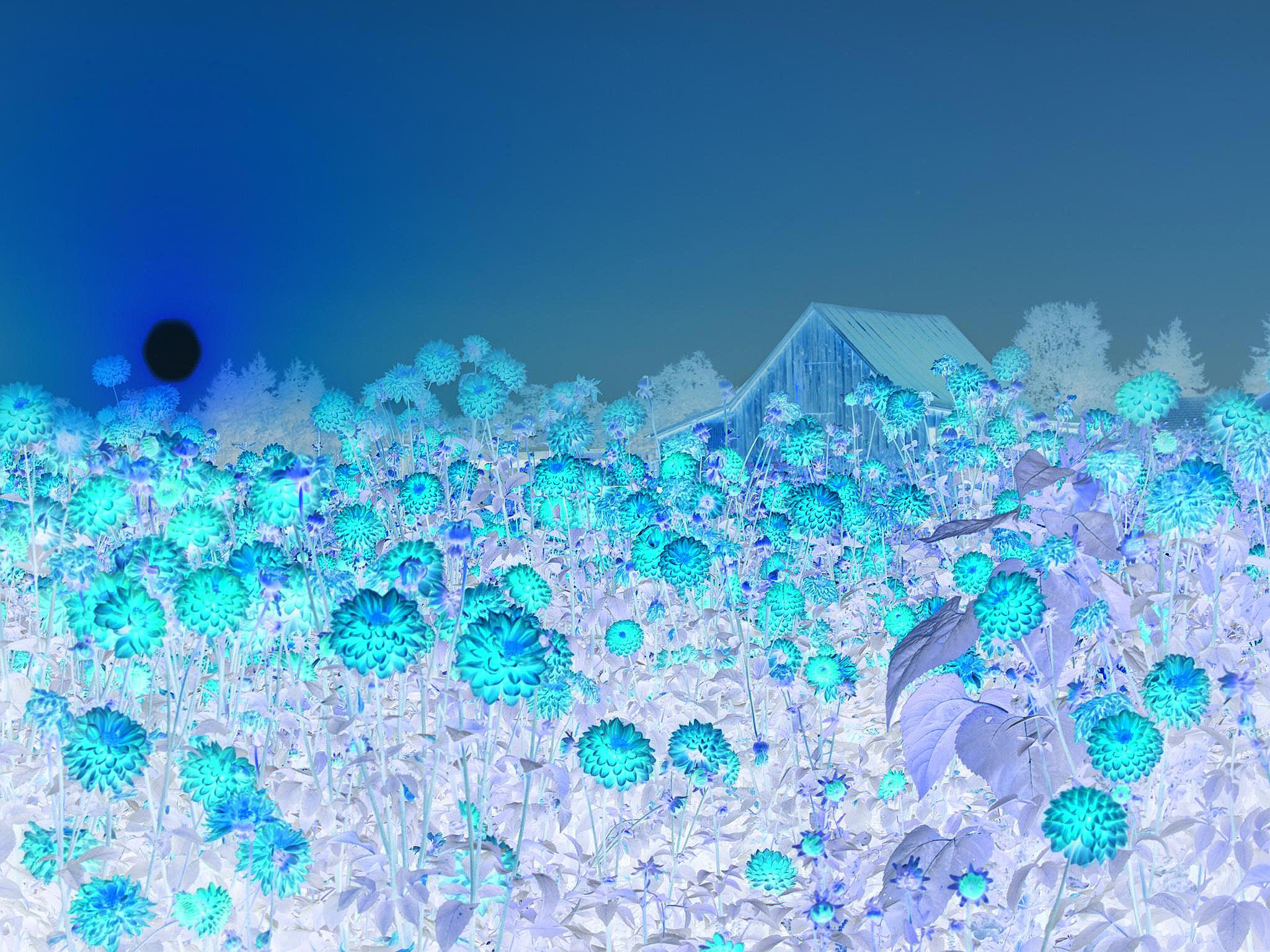


Рисунок 2.9 – Инвертированное и с увеличенном контрасте изображение разрешением с)

Результаты трёхкратного выполнения программ показаны в таблице 2.1:

Таблица 2.1:

|  |  |  |
| --- | --- | --- |
| Операция над изображением и тип использованного процессора | Среднее время выполнения отдельных операций, c | Среднее время для всех операция, c |
| Elapsed time (CPU) inversion 1024x768 | 0,000605 | 14,273695 |
| Elapsed time (CPU) sharpen 1024x768 | 14,27309 |
| Elapsed time (GPU) inversion 1024x768 | 0,000246 | 0,000412 |
| Elapsed time (GPU) sharpen 1024x768 | 0,000166 |
| Elapsed time (CPU) inversion 1280 x 960 | 0,000918 | 22,560168 |
| Elapsed time (CPU) sharpen 1280 x 960 | 22,55925 |
| Elapsed time (GPU) inversion 1280 x 960 | 0,000233 | 0,000469 |
| Elapsed time (GPU) sharpen 1280 x 960 | 0,000236 |
| Elapsed time (CPU) inversion 2048 x 1536 | 0,002083 | 48,882163 |
| Elapsed time (CPU) sharpen 2048 x 1536 | 48,88008 |
| Elapsed time (GPU) inversion 2048 x 1536 | 0,00014743 | 0,00037963 |
| Elapsed time (GPU) sharpen 2048 x 1536 | 0,0002322 |

# Вывод

В ходе выполнения лабораторной работы изучены инструменты многопоточности на GPU с использованием CUDA и написание программы для обработки изображений, а именно инвертирования и увеличения контрастности.

Исходя из результатов, продемонстрированных в таблице 2.1 можно сделать вывод, что использования CUDA значительно увеличивает скорость попиксельной обработки изображений различных разрешений, в 34 644 раз для изображения 1024x768, в 48 102 раз для изображения 1280 x 960 и в 128 762 раз для изображения 2048 x 1536.