**Лабораторная работа 4**

**Цель:** изучение основ работы с текстурной памятью на CUDA, реализация цифровой обработки сигналов на CUDA.

Задание

Дано двухмерное изображение. Написать программу, реализующую фильтр Box Blur, применительно к исходному изображению. Радиус фильтра – параметр программы.

Данные в программу



Данные из программы  


**Задание 2**: изучить основы работы с текстурным массивом на CUDA. Написать программу на Cи, реализующую цифровой билинейный фильтр для масштабирования изображения с использованием текстурного массива и CUDA runtime API.

Данные в программу



Данные из программы  


Поворот осуществлен на 45 градусов.

**Контрольные вопросы**

1. Где расположена текстурная память?

На графическом процессоре (GPU)

1. Основные особенности текстурной памяти?

Кеширование: Текстурная память поддерживает кеширование, что может повысить производительность при обращении к данным.

Фильтрация: Возможность автоматической фильтрации текстур для лучшей детализации и интерполяции.Где может располагаться текстура?

1. Где может располагаться текстура?

Текстурная память может располагаться как в оперативной памяти, так и в константной памяти устройства.

1. Основные функции работы с текстурным массивом?

tex1Dfetch, tex2D, tex3D: Функции для доступа к элементам текстурного массива в одномерных, двумерных и трехмерных случаях соответственно.

1. Что такое текстурная ссылка?

Текстурная ссылка - это особый тип переменной, предоставляемый CUDA, который связывается с текстурным массивом. Это позволяет использовать текстурную память для доступа к данным в ядрах CUDA.

1. Основные функции работы с текстурными ссылками?

cudaBindTexture, cudaBindTextureToArray: Функции для привязки текстурной ссылки к текстурному массиву.

cudaUnbindTexture: Функция для отвязки текстурной ссылки от текстурного массива.

1. Какова основная цель использования текстурной памяти при выполнении задач цифровой обработки сигналов?

Текстурная память эффективна при выполнении задач цифровой обработки сигналов, так как она обеспечивает быстрый доступ к данным с автоматической фильтрацией и кешированием, что показывает высокую производительность в тех случаях, когда данные обрабатываются в определенном порядке или требуется близкое расположение данных в памяти для улучшения кэш-попаданий.