

Отчет по лабораторной работе:

Анализ производительности параллельной обработки BMP-изображений

1. Цель работы

Целью данной работы является реализация параллельных версий алгоритмов для обработки изображений в формате BMP (поворот, размытие по Гауссу) и оценка прироста производительности по сравнению с последовательными аналогами.

2. Теоретическая часть

В ходе работы были применены следующие стратегии распараллеливания с использованием стандартной библиотеки C++ (`<thread>`):

- **Поворот изображения (rotate...):** Обработка изображения была разделена по строкам. Каждая строка (или группа строк) обрабатывалась в отдельном потоке, так как вычисление новой позиции для каждого пикселя является независимой задачей. Это классическая задача типа "embarrassingly parallel".
- **Размытие по Гауссу (applyGaussSeparable):** Алгоритм состоит из двух независимых проходов: горизонтального и вертикального. Каждый проход был распараллелен путем разделения изображения на горизонтальные полосы, которые обрабатывались независимыми потоками.
- **Операции ввода/вывода и вспомогательные функции:** Функции `loadBMPFile` и `saveBMPFile` были оставлены последовательными
- **Инструмент для замеров:** Для измерения времени выполнения использовалась библиотека `<chrono>` (`std::chrono::high_resolution_clock`). Каждый тест проводился 5 раз, для отчета было взято среднее арифметическое значение.

3. Результаты замеров (на фото 4к разрешения)

Функция	Последовательная версия (мс)	Параллельная версия (мс)	Ускорение (раз)
rotateBMP90Clockwise	526.17	200.97	2.6
rotateBMP90Counter-clockwise	852.52	207.85	4.11
applyGaussSeparable	6904.46	1987.19	3.47

Примечание: Ускорение = (Время последовательной версии) / (Время параллельной версии).

4. Выводы

В результате проделанной работы были успешно реализованы и протестированы параллельные алгоритмы обработки BMP-изображений. Экспериментальные замеры показали, что применение многопоточности позволяет существенно (в 3.4 раз) ускорить выполнение вычислительно-сложных операций. Наибольший прирост производительности был достигнут для функции ***applyGaussSeparable***. Было подтверждено, что для операций, ограниченных скоростью ввода-вывода, распараллеливание неэффективно.