



LABORATORIES

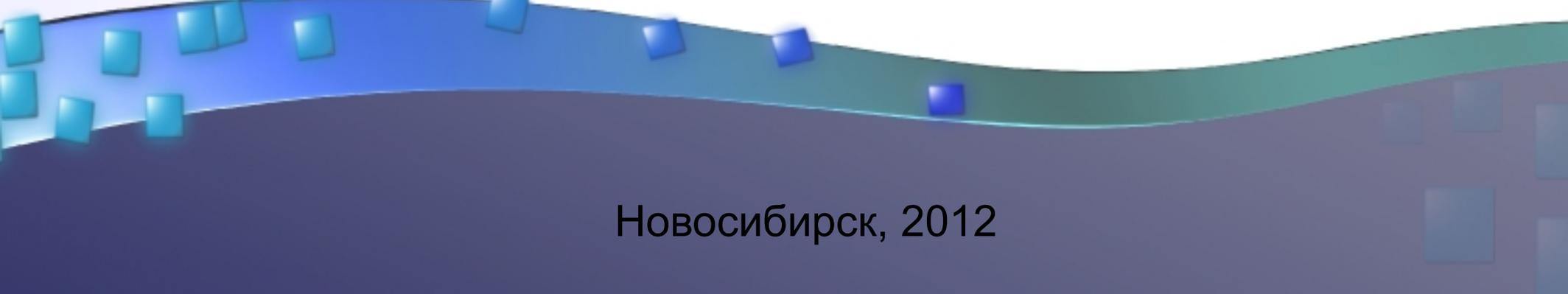
Опыт портирования среды для HDR-обработки изображений на GPU и APU

Авторы:

Кривов М.А.

Притула М.Н.

Елизаров С.Г.



Новосибирск, 2012

Содержание

- **Цель работ**
- Описание и анализ алгоритма
- Произведённые модификации
- Оценка производительности
- Выводы

Решаемая проблема

- Проблемы
 - Монитор может отобразить 8 бит на один цветовой канал (256 цветов).
 - Профессиональные фотокамеры сохраняют до 12 бит на один цветовой канал (4096)
 - А человеческий глаз воспринимает намного больше оттенков
- Результат
 - Фотографии нереалистичны

Решаемая проблема



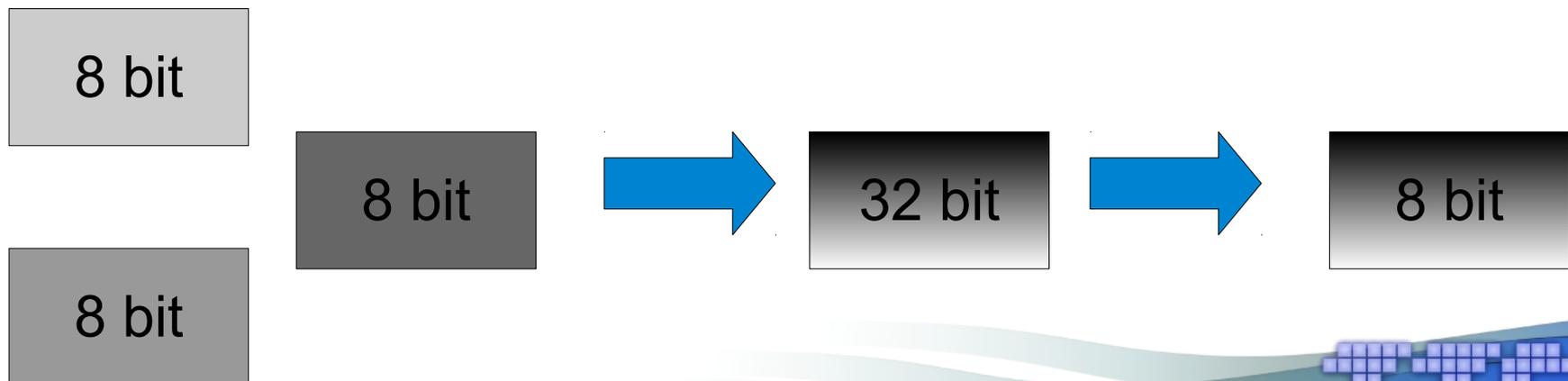
Мы никогда не узнаем,
что было за этой веткой



Из-за солнца невозможно
что-либо разобрать

HDR-преобразование

- Шаг 1. Делается несколько фотографий с разной засветкой
- Шаг 2. По ним создаётся одна HDR-фотография с 32 битами на канал.
- Шаг 3. С помощью алгоритма Tone Mapping HDR-фотография переводится в обычную



HDR-преобразование



Хоть всё равно и не видно, но там МГУ им. М.В. Ломоносова



Объекты становятся более понятными

Демонстрация пакета LuminanceHDR

Содержание

- Цель работ
- **Описание и анализ алгоритма**
- Произведённые модификации
- Оценка производительности
- Выводы

Алгоритм Mantiuk06

- Шаг 1. Выделение контраста

$$Contrast_i(x, y) = \log\left(\frac{Img(x + dx_i, y + dy_i).r + Img(x + dx_i, y + dy_i).g + Img(x + dx_i, y + dy_i).b}{Img(x, y).r + Img(x, y).g + Img(x, y).b}\right)$$

- Шаг 2. Обработка контраста

$$NewContrast_i(x, y) = grad^{-1} f(grad Contrast_i(x, y))$$

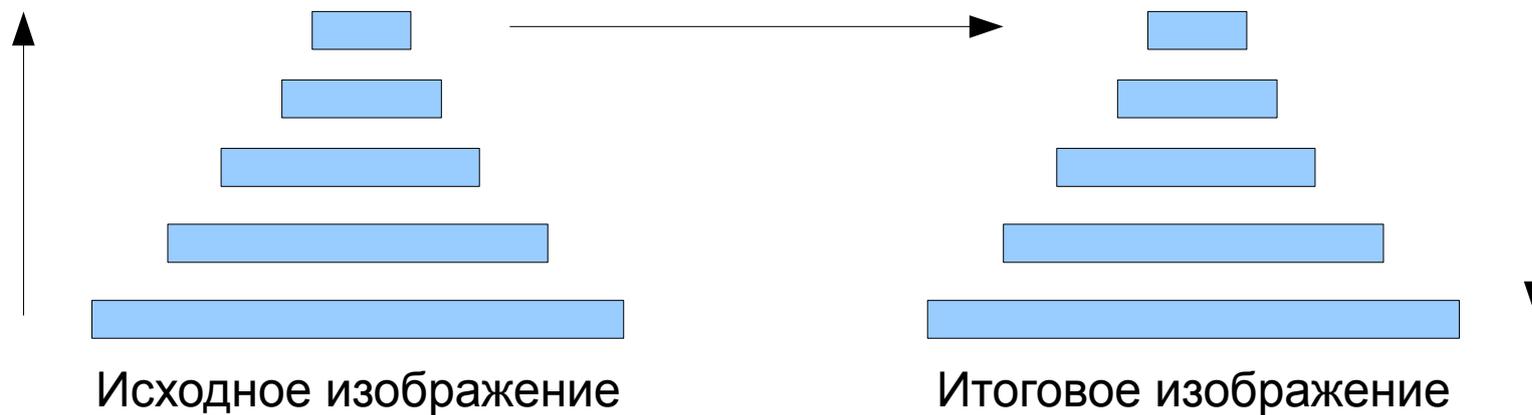
- Шаг 3. Наложение контраста

$$J(NewImg, NewContrast_i) \rightarrow 0$$

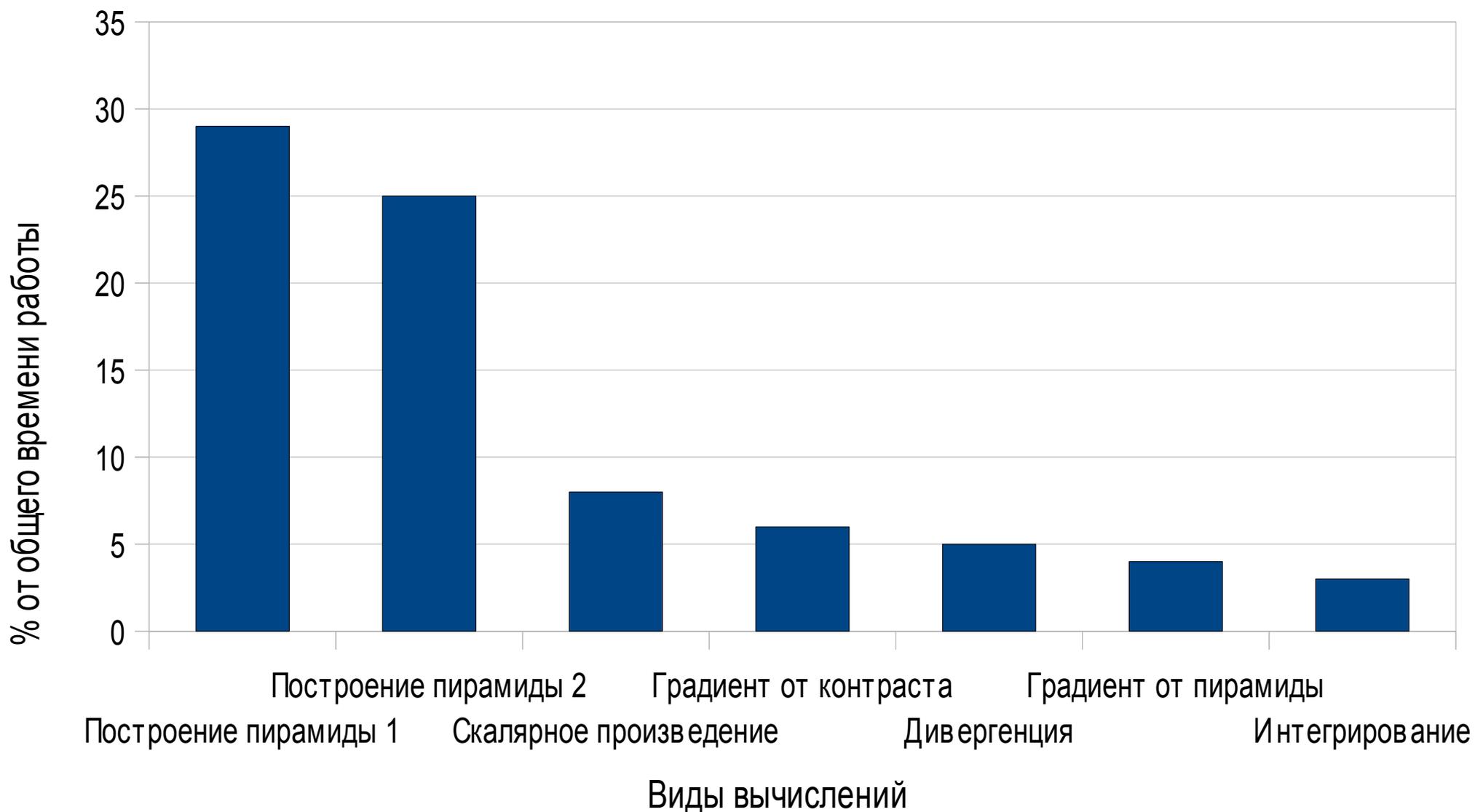
dx, dy, f, J, N — параметры конкретной реализации

Минимизация функционала

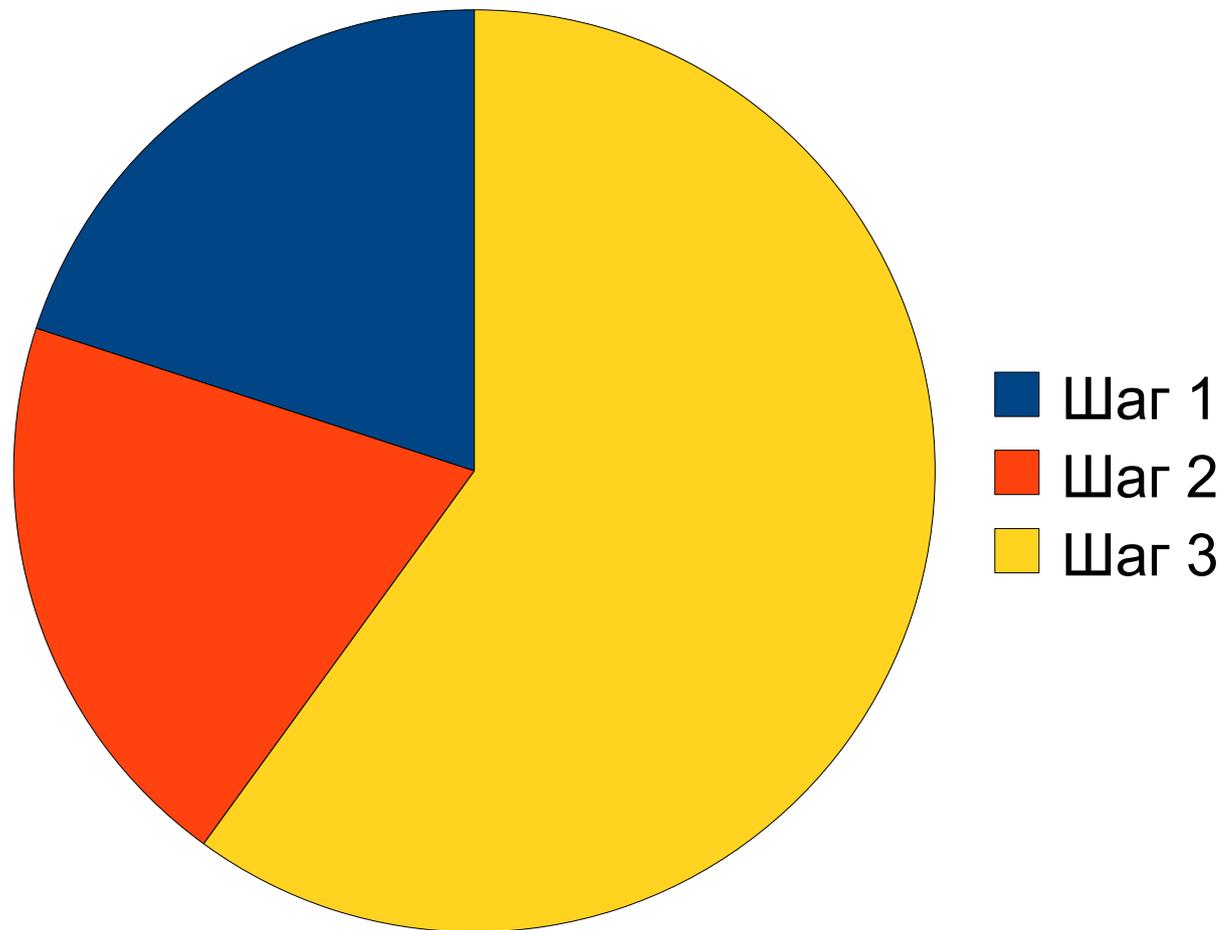
- Фактически требуется решить аналог дифференциального уравнения
- Используется итерационный метод решения



Времена работы



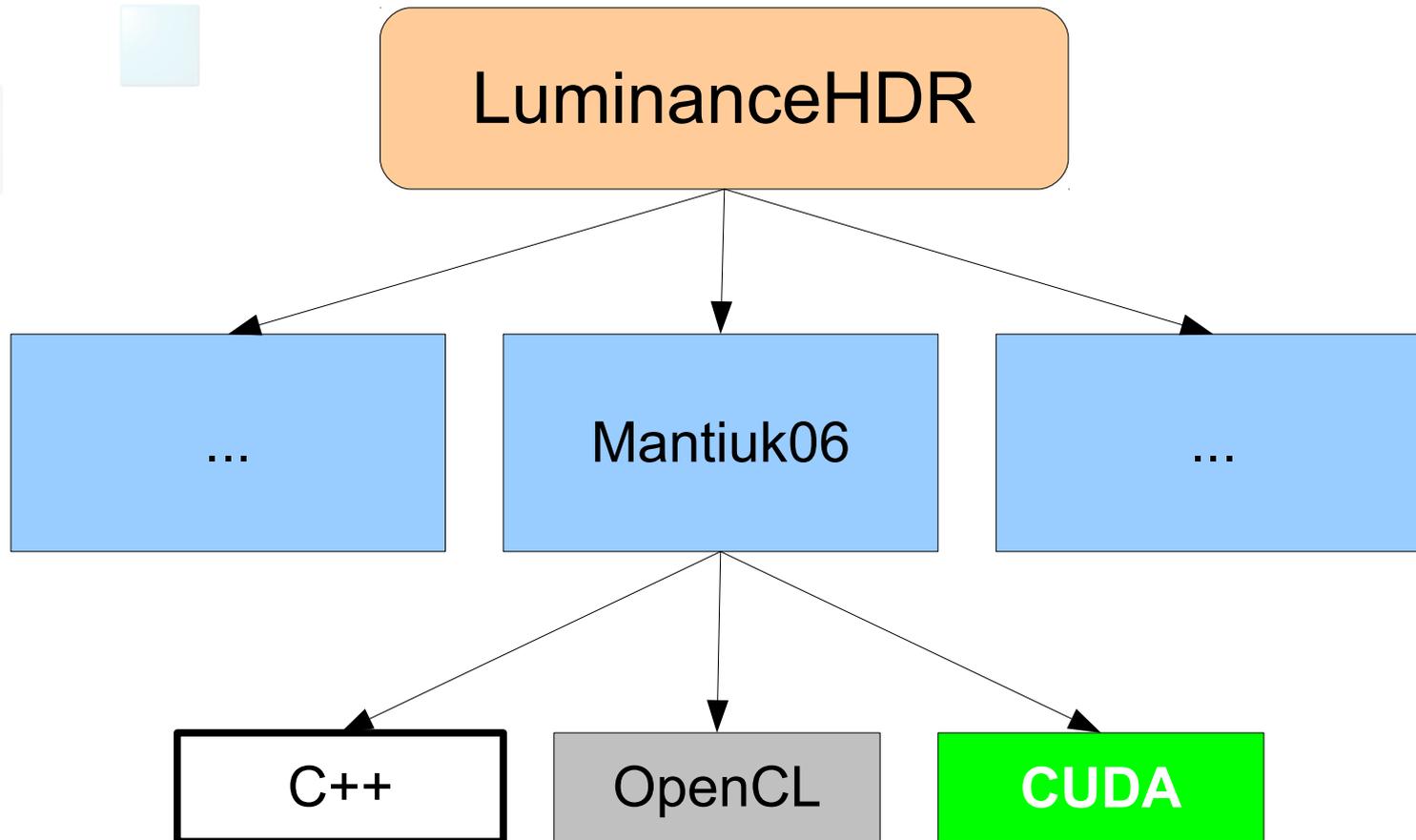
Времена работы



Содержание

- Цель работ
- Описание и анализ алгоритма
- **Произведённые модификации**
- Оценка производительности
- Выводы

Архитектура



Общие оптимизации

- Оптимизация работы с памятью
 - Отказ от new/delete в пользу буферов памяти
- Изменение пропорций пирамид
 - Исходный размер выбирался как число вида $2^n * m$, где $m < 7$
- Оптимизация информационного графа
 - Удалось отказаться от ряда промежуточных массивов

Специфичные для GPU ОПТИМИЗАЦИИ

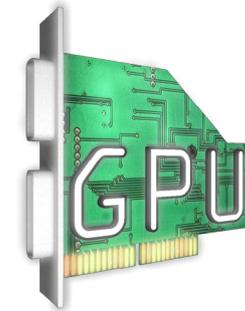
- Подбор оптимально топологии данных (grid)
Ускорение на 20-30%
- Использование shared/local памяти
Сокращение обращений памяти в 2-4 раза
- Использование аппаратной интерполяции
«Бесплатная» возможность масштабирования

Содержание

- Цель работ
- Описание и анализ алгоритма
- Произведённые модификации
- **Оценка производительности**
- Выводы

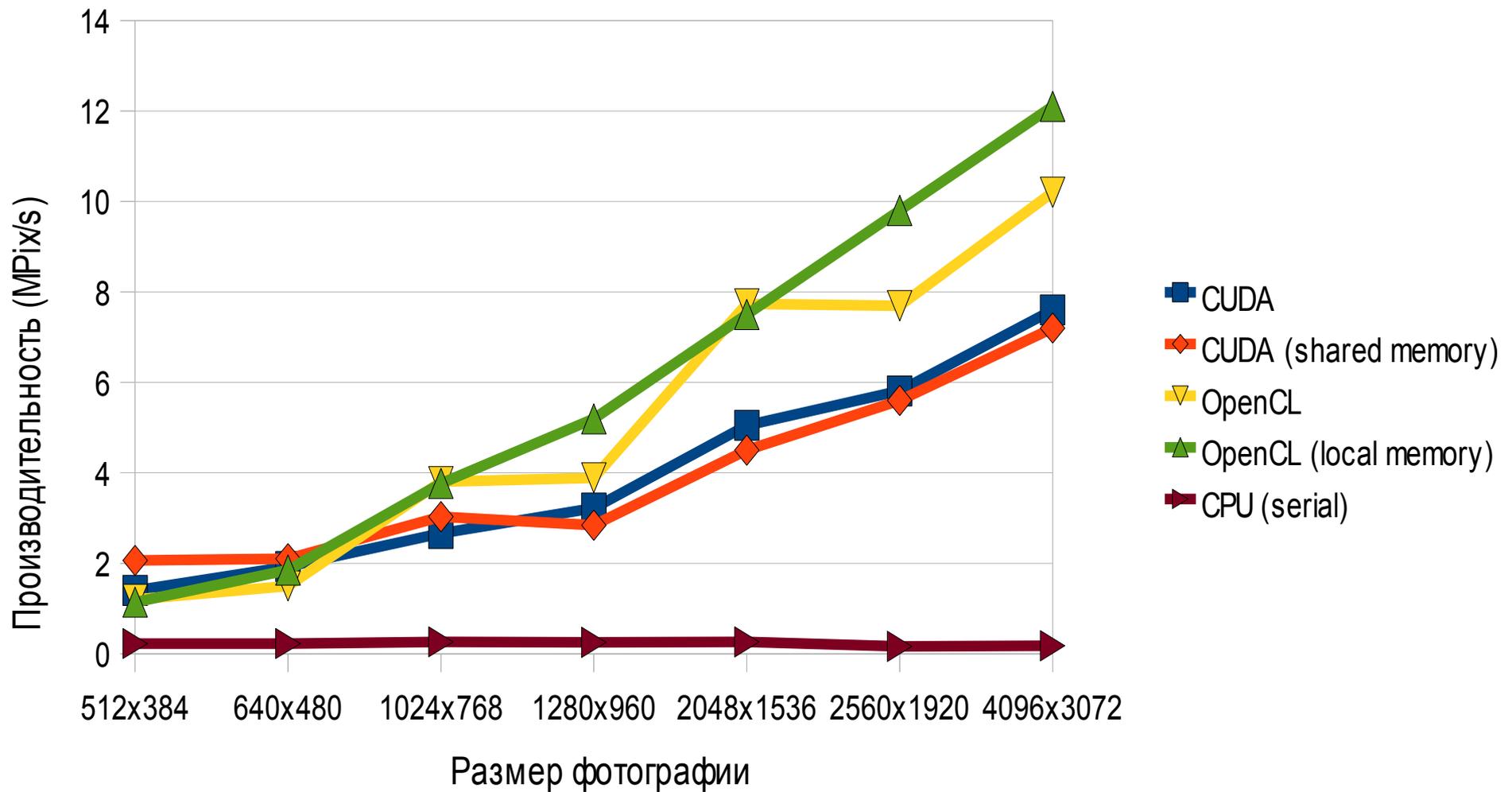
Тестирование GPU

- Система #1
 - **NVidia** GeForce 580 GTX
 - 512 CUDA-ядер
 - 1.5 TFlops
 - **Intel** Xeon E3 1230
- Система #2
 - **NVidia** GeForce 555M
 - 144 CUDA-ядер
 - 340 GFlops
 - **Intel** Core i7 2670



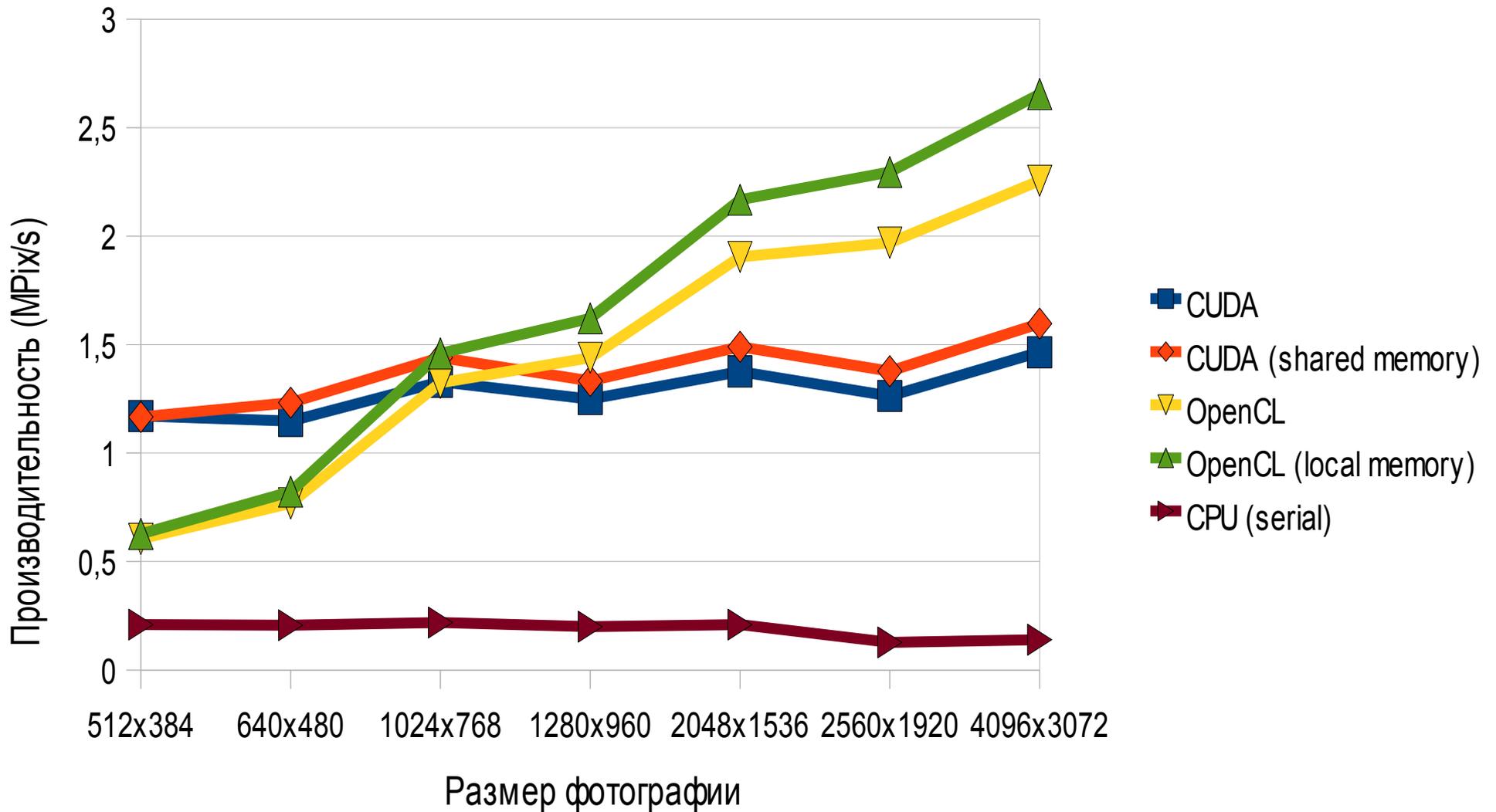
Результаты

GeForce 580 GTX + Xeon E3 1230



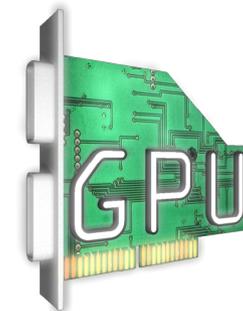
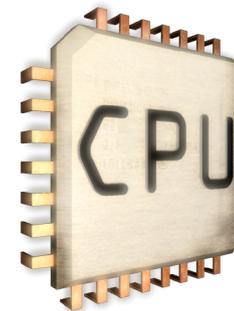
Результаты

GeForce 555M + Core i7 2670



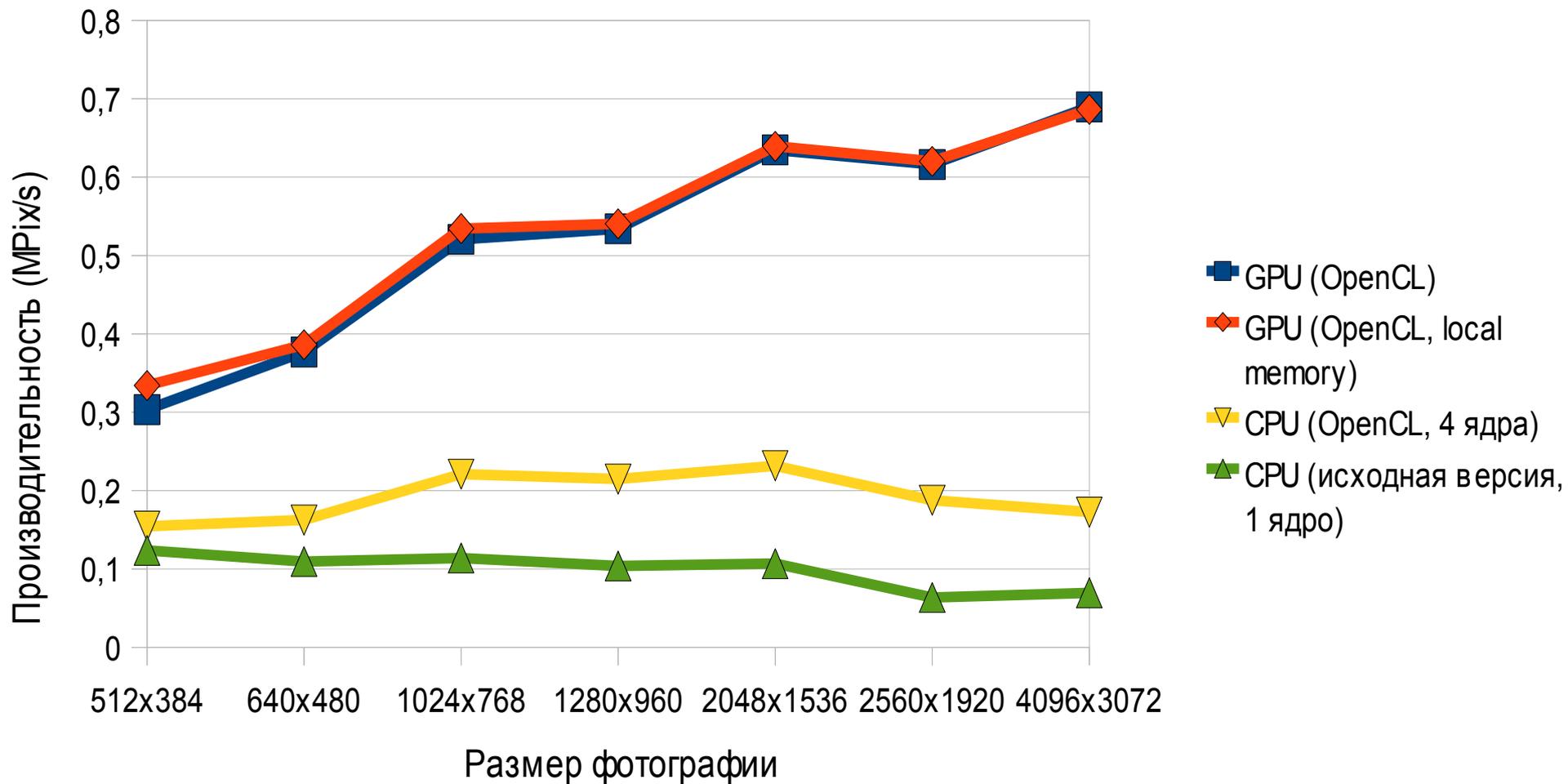
Тестирование APU

- AMD APU A8-3850
 - 4 CPU-ядра @ 2.9 GHz
 - 400 GPU ядер @ 500 MHz
 - Суммарно 0.5 TFlops
- AMD APU E-350
 - 2 CPU-ядра @ 1.6 GHz
 - 80 GPU ядер @ 400 Mhz
 - Суммарно 0.1 TFlops



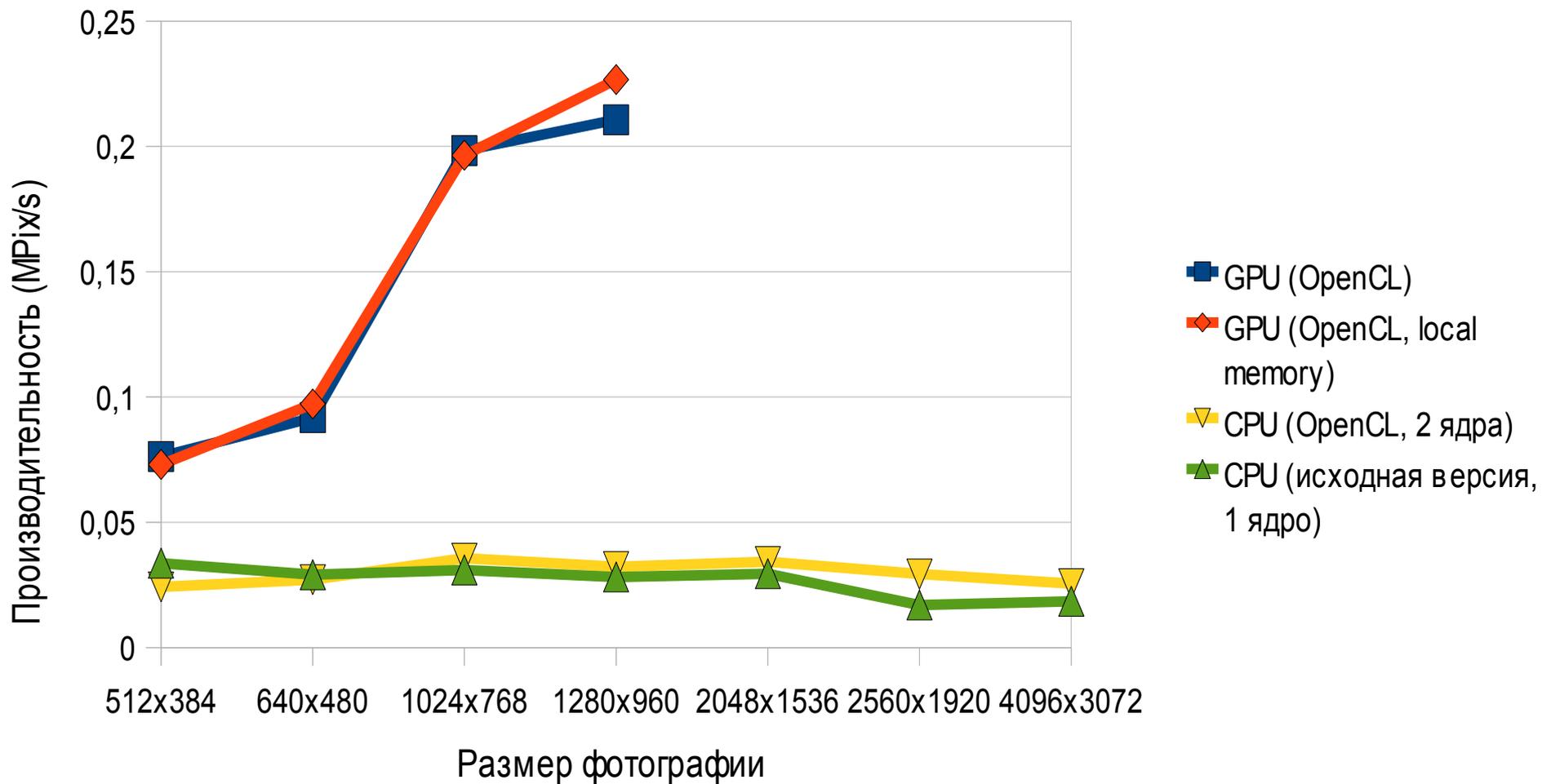
Результаты

AMD APU A8-3850



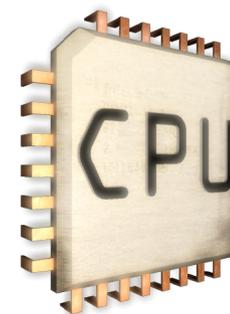
Результаты

AMD APU E-350



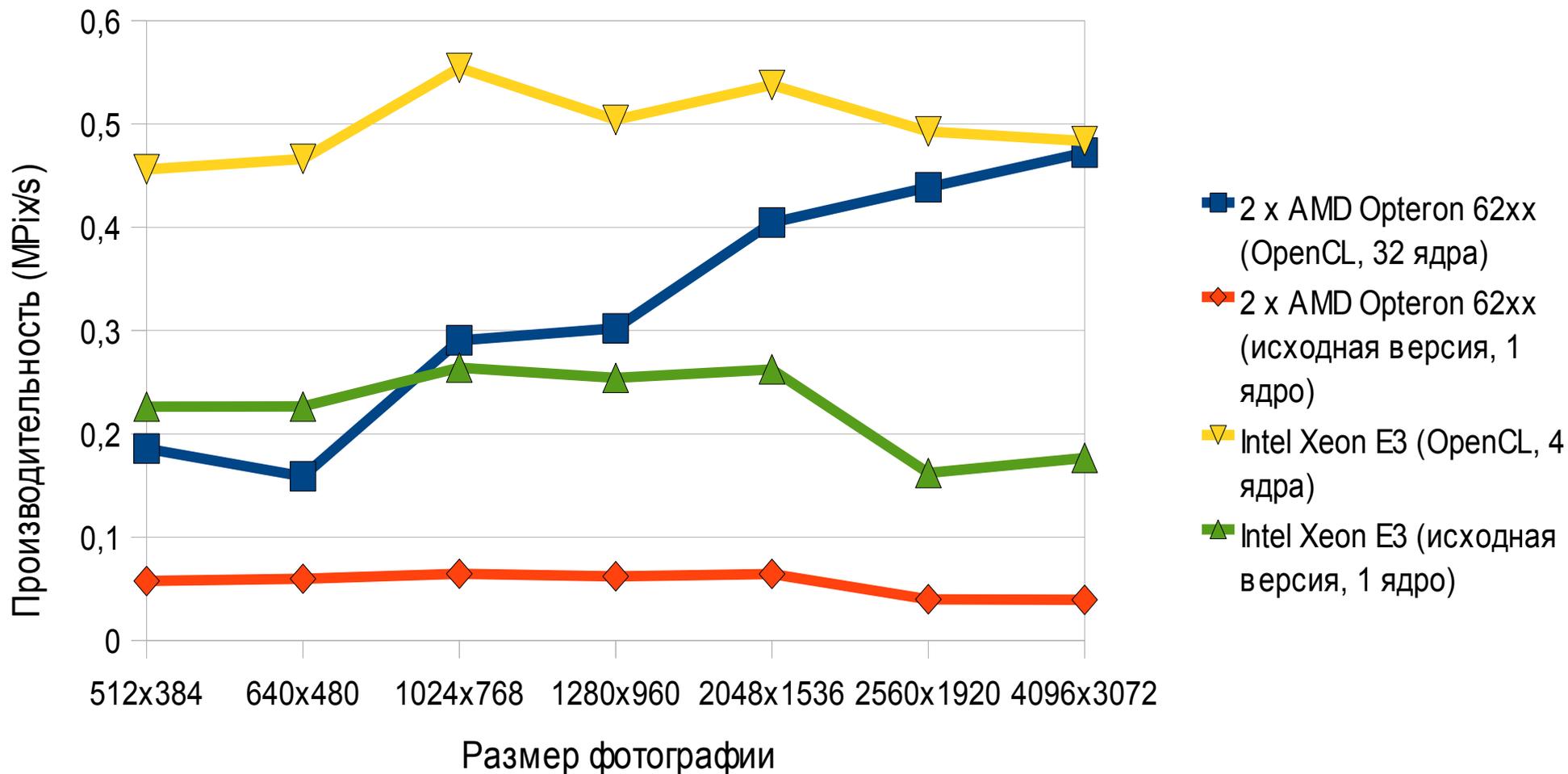
Тестирование CPU

- Intel Xeon E3 1230
 - 4/8 ядер @ 3.2 GHz
 - 100/50 GFlops
 - Intel OpenCL driver
- 2 x AMD Opteron 62xx
 - 2x16 ядер @ 2 GHz
 - 260/130 GFlops
 - AMD OpenCL driver



Результаты

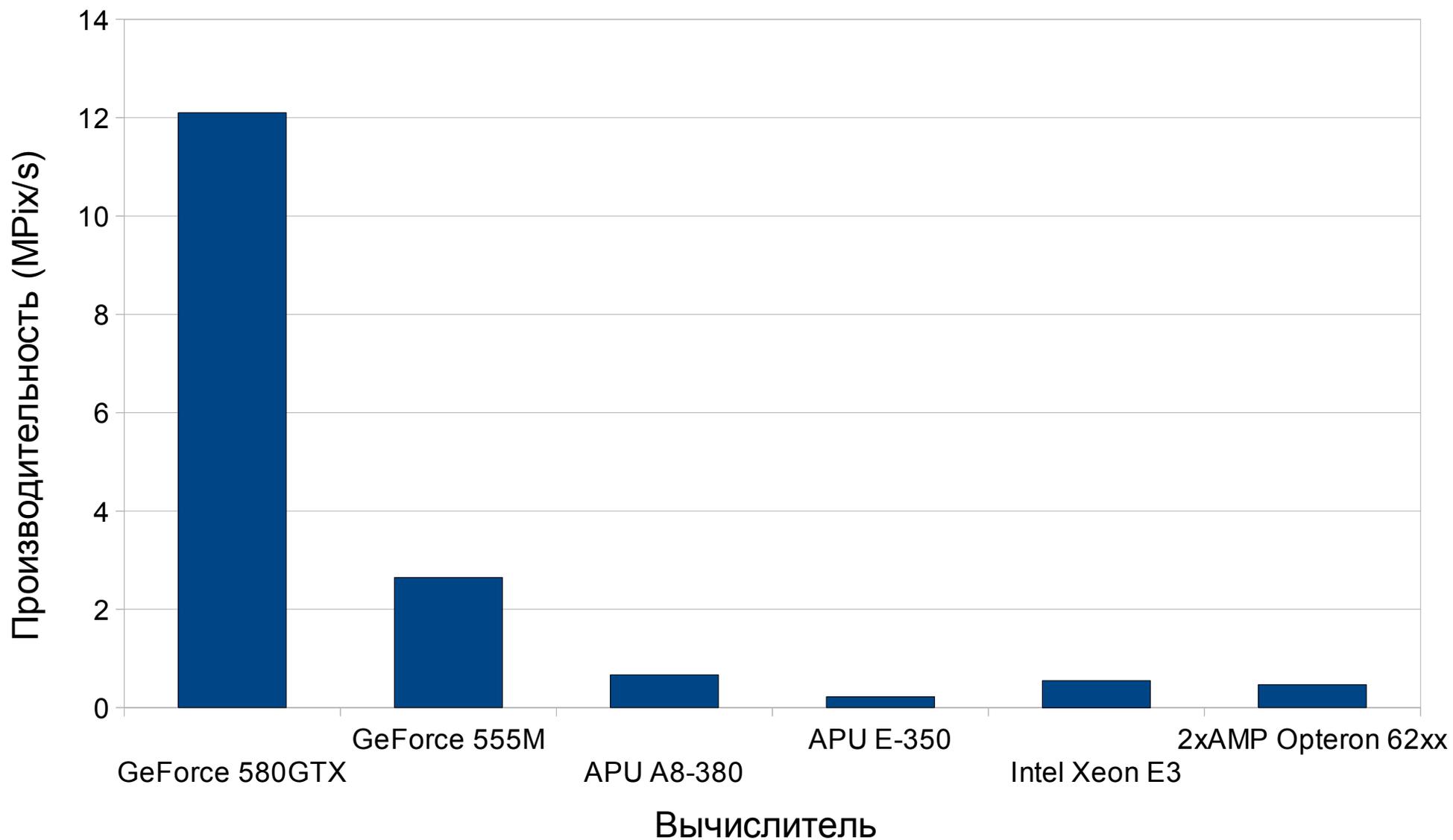
Intel Xeon E3-1230 vs 2xAMD Opteron 62xx



Содержание

- Цель работ
- Описание и анализ алгоритма
- Произведённые модификации
- Оценка производительности
- **Выводы**

Производительность всех систем





Вопросы?

(m_krivov@ttgLabs.com)