Практическое задание №1 Реализация многопоточных вычислений на CPU

Написать 2 программы на одном из языков программирования (С/ C++, C#, Java, Python) с использованием инструментов многопоточной обработки на базе CPU.

Содержание отчета.

1. Титульный лист
2. Задача по полученному варианту.

(если последняя цифра номера зачетной книжки/ списка внутри группы/ дня рождения 0 или 5: вариант 1;

если последняя цифра номера зачетной книжки/ списка внутри группы/ дня рождения 1 или 6: вариант 2;

если последняя цифра номера зачетной книжки/ списка внутри группы/ дня рождения 2 или 7: вариант 3;

если последняя цифра номера зачетной книжки/ списка внутри группы/ дня рождения 3 или 8: вариант 4;

если последняя цифра номера зачетной книжки/ списка внутри группы/ дня рождения 4 или 9: вариант 5

1. Листинг программ
2. Примеры входных и выходных файлов для программ A и B
3. Описание результатов тестирования
4. Выводы

**Вариант 1**

**Программа A**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при пятикратном перезапуске программы.

Загрузить цветное изображение.

Получить значения интенсивности ,

где – интенсивность пикселя *v*, – значение красной компоненты пикселя , – значение зелёной компоненты пикселя , – значение синей компоненты пикселя .

Установить значение скалярной величины - порога Threshold (любое значение от 100 до 200). Те значения интенсивности, которые меньше порогового значения установить в 0, а которые больше Threshold установить в 1.

Выполнить операцию эрозии [https://habr.com/ru/post/113626/ или https://intuit.ru/studies/courses/10621/1105/lecture/17989?page=4] над полученной матрицей из 0 и 1. Примечание: Нужно задать шаг эрозии (любое значение от 1, 2 или 3). Получить изображение из результата путем установки вместо значений 0 – пикселей черного цвета (0, 0, 0), и вместо значений 1 – пикселей белого цвета. Сохранить результат в файл.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Программа B**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы.

Загрузить цветное изображение.

Установить шаг переноса по оси Х и оси Y [https://habr.com/ru/post/113626/]. Выполнить сдвиг всех пикселей на нужный шаг. Пограничные пиксели, содержавшие прежние пиксели, заменить цветом RGB = (187, 38, 73). Выполнить свертку с фильтром, добавляющим размытие, 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.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Вариант 2**

**Программа A**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при пятикратном перезапуске программы.

Загрузить цветное изображение.

Установить шаг переноса по оси Х и оси Y [https://habr.com/ru/post/113626/]. Выполнить сдвиг всех пикселей на нужный шаг. Пограничные пиксели, содержавшие прежние пиксели, заменить цветом RGB = (187, 38, 73). Выполнить свертку с фильтром, добавляющим размытие, 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.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Программа B**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы.

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.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Вариант 3**

**Программа A**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при пятикратном перезапуске программы.

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.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Программа B**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы.

Загрузить цветное изображение.

Получить значения интенсивности ,

где – интенсивность пикселя *v*, – значение красной компоненты пикселя , – значение зелёной компоненты пикселя , – значение синей компоненты пикселя .

Установить значение скалярной величины - порога Threshold (любое число от 1 до 255). Те значения интенсивности, которые меньше порогового значения установить в 0, а которые больше Threshold установить в 1.

Выполнить операцию наращивания (диляции/ дилатации) [https://habr.com/ru/post/113626/ или https://intuit.ru/studies/courses/10621/1105/lecture/17989?page=4] над полученной матрицей из 0 и 1. Примечание: Нужно задать шаг наращивания (любое значение от 1, 2 или 3). Получить изображение из результата путем установки вместо значений 0 – пикселей черного цвета (0, 0, 0), и вместо значений 1 – пикселей белого цвета. Сохранить результат в файл.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Вариант 4**

**Программа A**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при пятикратном перезапуске программы.

Загрузить цветное изображение.

Выполнить свертку с фильтром, придающим рельеф с каждым из цветовых каналов изображения. Ядро преобразования можно найти по ссылке: [https://docs.gimp.org/2.8/ru/plug-in-convmatrix.html]. Уменьшить масштаб изображения в два раза: область размера 2 x 2 пикселей заменить пикселем, цветовые компоненты которого рассчитываются путем усреднения цветовых компонент указанной области. Сохранить результат в файл.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Программа B**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы.

Получить значения интенсивности ,

где – интенсивность пикселя *v*, – значение красной компоненты пикселя , – значение зелёной компоненты пикселя , – значение синей компоненты пикселя .

Установить значение скалярной величины - порога Threshold (любое значение от 100 до 200). Те значения интенсивности, которые меньше порогового значения установить в 0, а которые больше Threshold установить в 1.

Выполнить операцию эрозии [https://habr.com/ru/post/113626/ или https://intuit.ru/studies/courses/10621/1105/lecture/17989?page=4] над полученной матрицей из 0 и 1. Примечание: Нужно задать шаг эрозии (любое значение от 1, 2 или 3). Получить изображение из результата путем установки вместо значений 0 – пикселей черного цвета (0, 0, 0), и вместо значений 1 – пикселей белого цвета. Сохранить результат в файл.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Вариант 5**

**Программа A**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при пятикратном перезапуске программы.

Загрузить цветное изображение.

Получить значения интенсивности ,

где – интенсивность пикселя *v*, – значение красной компоненты пикселя , – значение зелёной компоненты пикселя , – значение синей компоненты пикселя .

Установить значение скалярной величины - порога Threshold (любое число от 1 до 255). Те значения интенсивности, которые меньше порогового значения установить в 0, а которые больше Threshold установить в 1.

Выполнить операцию наращивания (диляции/ дилатации) [https://habr.com/ru/post/113626/ или https://intuit.ru/studies/courses/10621/1105/lecture/17989?page=4] над полученной матрицей из 0 и 1. Примечание: Нужно задать шаг наращивания (любое значение от 1, 2 или 3). Получить изображение из результата путем установки вместо значений 0 – пикселей черного цвета (0, 0, 0), и вместо значений 1 – пикселей белого цвета. Сохранить результат в файл.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.

**Программа B**

Произвести тестирование программы для разных размеров обрабатываемых файлов изображений. Для тестирования взять файлы размерами: 10240 x 7680, 12800 x 9600, 20480 x 15360. Получить среднее значение работы процедуры обработки каждого изображения при троекратном перезапуске программы.

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.

Произвести оценку производительности алгоритма обработки указанных изображений для 2, 4, 6, 8, 10, 12, 14 и 16 потоков на CPU.