

# Задание на зачет

## Требования

Написать приложение для обработки изображения с использованием масочного(матричного) фильтра.

Ввод исходных данных:

- размер матрицы фильтра(маски)  $N$  и значения коэффициентов
- загрузка изображения

Пользовательский интерфейс:

- используется QML. Отображается два изображения (исходное и после обработки). Управление через главное меню приложения и кнопки.
- Сохранение обработанного изображения в файл.
- отображение прогресса обработки изображения

Расчетная многопоточная часть:

- QML плагин на C++ (QML extension plugin)
- обработка граничных пикселей - методом дополнения (создается временное промежуточное изображение размерами  $(width + 2 * N/2, height + 2 * N/2)$ . В центр изображения копируется исходная картинка, а края заполняются крайними пикселями исходного изображения. Фильтр применяется к промежуточному изображению, а потом из него извлекается результат)
- использовать многопоточный механизм Qt Concurrent map-reduce для фоновой обработки изображения
- запрограммировать фильтр:
  - 501 - размытие
  - 511 - медианный

Во время работы:

- отображать прогресс фильтрации (процент уже обработанных пикселей исходного изображения)
- иметь возможность прервать фильтрацию

Результат:

- вывести время фильтрации
- отобразить отфильтрованное изображение (показать рядом с оригиналом в интерфейсе)
- возможность сохранить отфильтрованное изображение в файл jpg (или другой).

# Теоретические сведения

Одним из методов предварительной обработки изображений является масочная фильтрация, когда по изображению перемещается скользящее окно с заданными в нем весовыми коэффициентами (маска), обработка ведется по сравнительно небольшой окрестности каждой точки изображения. Достоинством этого метода является хорошее быстродействие.

Масочная фильтрация используется для получения различных эффектов, таких как подавление шумов, выделение контурных линий, уменьшение резкости, усиление резкости, выполнение «тиснения» и других.

Преобразование изображения в точке  $(i, j)$  имеет вид

$$g_{ij} = \sum_D a_{kl} f_{i+k, j+l}$$

суммирование ведется по некоторой окрестности  $D$  точки. Значение характеристики пикселя (это может быть компонент цвета, яркость) в этой точке и в ее окрестности умножаются на коэффициенты  $a_{kl}$ , преобразованное значение  $(i, j)$ -го пикселя есть сумма этих произведений. Обычно набор коэффициентов  $a_{kl}$  представляют в виде прямоугольной матрицы (маски) нечетной размерности, например размерности  $3 \times 3$ :

$$H = 1/K \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}.$$

Фильтрация осуществляется последовательным перемещением маски по всем пикселям исходного изображения. При каждом положении маски производятся упомянутые выше операции, а именно перемножение весовых множителей  $a_{kl}$  с соответствующими значениями пикселя исходного изображения и суммированием произведений (произведение матриц). Полученное значение делится на нормирующий множитель  $K$  для того чтобы не происходило изменения среднего значения пикселя обработанного изображения, необходимо задать  $K$  следующим образом:

$$K = \sum_{i,j} a_{ij}$$

С алгоритмической точки зрения, получение этих эффектов заключается в вычислении свертки изображения с маской. Например матрица маски размером  $3 \times 3$  содержит три строки по три числа в каждой. Чтобы преобразовать один пиксель в изображении (назовем его центральным), матрица маски «умножается» на матрицу значений пикселей исходного изображения такой же размерности вокруг центрального пикселя, в результате получается новое значение центрального пикселя. Этот процесс повторяется для каждого пикселя в изображении, тем самым изображение фильтруется. Коэффициенты ядра определяют результат процесса фильтрации. Ядро размытия, например, состоит из совокупности коэффициентов, каждый из которых меньше 1, а их сумма составляет 1. Это означает, что каждый пиксель поглотит что-то из яркости соседей, но полная яркость изображения останется неизменной. Если сумма коэффициентов больше чем 1, яркость увеличивается; если меньше чем 1, яркость уменьшится.

При подготовке к размыванию цифровое изображение считывается в память компьютера в виде красной, зеленой и синей компонент цвета каждого пикселя. Ядро размывания заданного размера  $N \times N$  применяется к красной, зеленой и синей компонентам цвета каждого пикселя в изображении в отдельности, результат суммируется. Итоговое изображение получается размытым по сравнению с оригиналом потому, что цвет каждого пикселя распространился среди соседей. Степень размывания можно увеличить, либо используя ядро большего размера, чтобы распределить цвета среди большего числа соседей, либо подбирая коэффициенты ядра и уменьшая влияние центрального коэффициента, либо фильтруя изображение еще раз с ядром размывания.

В ядре увеличения резкости центральный коэффициент больше 1, но окружен отрицательными числами, сумма которых на единицу меньше центрального коэффициента. Таким образом, увеличивается любой существующий контраст между цветом пикселя и цветами его соседей. Процесс увеличения резкости просто повысил существующий контраст между пикселями. При повторной обработке изображения четкость может увеличиться еще больше.

# Оценка

Максимальная оценка 50 баллов. Выставляется за выполнение работы в течении контрольного срока. При сдаче работы после контрольного срока применяется коэффициент 0.8. Проверка и оценка работы программы производится по пунктам:

1. Пользовательский интерфейс главного окна приложения написан на QML, расчетная часть - плагин для QML на C++
2. Элементы интерфейса правильно ведут себя при изменении размеров главного окна и заполняют всю его область
3. Используется многопоточность (map-reduce производится одновременным заданным количеством потоков)
4. в статусной строке отображается процесс выполнения фильтрации (процент обработанных пикселей)
5. по окончании фильтрации отображается обработанное изображение рядом с исходным (для сравнения)
6. в случае прерывания процесса фильтрации отображается промежуточный результат (то что успело обработаться)
7. обработанное изображение корректно сохраняется в файл (возможно просмотреть другими приложениями установленными в ОС)

Стоимость п.1 - 10 баллов, пп.2-7 - 5 баллов. Итоговая оценка получается вычитанием стоимости невыполненных пунктов задания из 50 баллов.