

## Реализация фоторедактора

В рамках данной работы необходимо реализовать программу, позволяющую открывать, сохранять и обрабатывать изображения в формате bmp24 с помощью следующих операций:

1. Перевод изображения в градацию серого
2. Оператор Собеля (градиент изображения)
3. Оператор Лапласа (вторая производная изображения)
4. Размытие по Гауссу
5. Выделение границ (опционально, используя оператор [Канни](#))
6. Обработка произвольно заданным ядром
7. Опционально медианный фильтр
8. Опционально графический интерфейс

Для консольной версии приложения предусмотреть аргументы командной строки для ввода имени входного и выходного файлов, а также применяемых операций.

## Формат bmp

Данные в формате BMP состоят из трёх основных блоков различного размера:

1. Заголовок из структуры BITMAPFILEHEADER и блока BITMAPINFO. Последний содержит:
  - а. информационные поля;
  - б. пиксельные данные.

При хранении в файле все заголовки идут с самого первого байта. Пиксельные данные могут находиться на произвольной позиции в файле (она указывается в поле OffBits структуры BITMAPFILEHEADER), в том числе и в удалении от заголовков.

### Заголовок [BITMAPFILEHEADER](#)

Поз. (hex)	Размер (байты)	Имя	Описание
00	2	Type	Сигнатура формата = BM
02	4	FileSize	Размер файла в байтах
06	4	Reserved	Зарезервированы и должны содержать ноль
0A	4	OffBits	Положение пиксельных данных относительно начала данной структуры (в байтах)

### Заголовок [BITMAPINFO](#)

Поз. (hex)	Размер (байты)	Имя	Описание
0E	4	bSize	Размер данной структуры в байтах

12	4	Width	Ширина растра в пикселях. Указывается целым числом со знаком. Ноль и отрицательные не документированы.
16	4	Height	Целое число со знаком, содержащее два параметра: высота растра в пикселях (абсолютное значение числа) и порядок следования строк в двумерных массивах (знак числа). Нулевое значение не документировано.
1A	2	Planes	В BMP допустимо только значение 1.
1C	2	BitCount	Количество бит на пиксель
1E	4	Compression	Указывает на способ хранения пикселей. 0 без сжатия
22	4	SizeImage	Размер пиксельных данных в байтах. Может быть обнулено, если хранение осуществляется без сжатия
26	4	XPelsPerMeter	Количество пикселей на метр по горизонтали и вертикали
2A	4	YPelsPerMeter	
2E	4	ClrUsed	Размер таблицы цветов в ячейках.
32	4	ClrImportant	Количество ячеек от начала таблицы цветов до последней используемой (включая её саму).

В рамках работы требуется обработка 24-битных bmp файлов без сжатия. В данном формате пиксели растра записываются однопиксельными горизонтальными полосками (строками). В памяти эти ряды записываются по порядку, но при положительном Height: начиная с самого нижнего, а при отрицательном: с самого верхнего. Внутри каждого горизонтального ряда пиксели записываются строго от левого к правому в формате bgra (Blue, Green, Red и опционально альфа-канал). Каждая строка дополняется нулями до кратного четырём байтам размера.

## Grayscale

Пусть  $A$  – исходное изображение, тогда изображение можно перевести в оттенки серого с помощью среднего арифметического всех компонент цвета:

$$G = \frac{A_r + A_g + A_b}{3}$$

где  $A_s$  – матрица, отвечающая за s-компоненту цвета

Второй способ перевода в градацию серого основан на вычислении яркости изображения:

$$G = K_r A_r + (1 - K_r - K_b) A_g + K_b A_b$$

где  $K_r = 0.299$ ,  $K_b = 0.114$  коэффициенты перевода, определяемые по особенностям человеческого восприятия цвета.

## Свёртка

Свёртка – это операция вычисления нового значения выбранного пикселя, учитывающая значения окружающих его пикселей. Для вычисления значения используется матрица, называемая ядром свертки. Обычно ядро свертки является квадратной матрицей  $n * n$ , где  $n$  — нечетное.

Во время вычисления нового значения выбранного пикселя ядро свертки как бы «прикладывается» своим якорем к данному пикселю. Окружающие пиксели так же накрываются ядром. Далее высчитывается сумма, где слагаемыми являются произведения значений пикселей на значения ячейки ядра, накрывшей данный пиксель. Сумма делится на сумму всех элементов ядра свертки. Полученное значение является новым значением выбранного пикселя. Если применить свертку к каждому пикселю изображения, то в результате получится некий эффект, зависящий от выбранного ядра свертки.

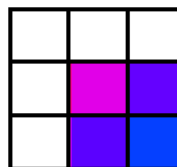
Пример вычисления свёртки для элемента  $a_{11}$ :

$$a_{11} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \circ \begin{bmatrix} 1 & 6 & 11 & 16 & 21 \\ 2 & 7 & 12 & 17 & 22 \\ 3 & 8 & 13 & 18 & 23 \\ 4 & 9 & 14 & 19 & 24 \\ 5 & 10 & 15 & 20 & 25 \end{bmatrix} =$$

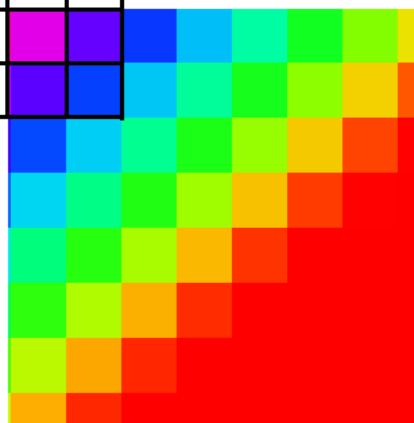
$$(1 * 1 + 6 * 0 + 11 * 0 + 2 * 0 + 7 * 1 + 12 * 0 + 3 * 0 + 8 * 0 + 13 * 1) / 3 = 7$$

При вычислении краевых пикселей возникает проблема отсутствия соседей у вычисляемого пикселя.

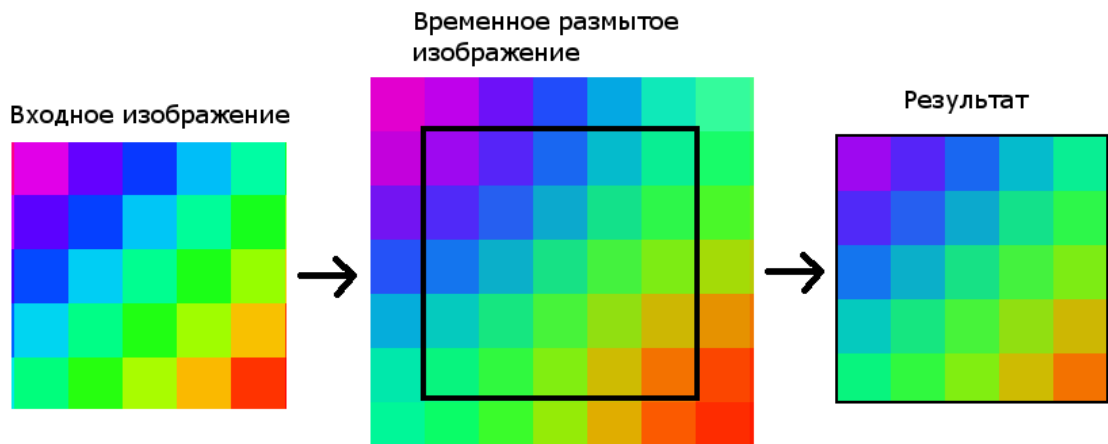
Матрица



Изображение



Для решения данной проблемы пикселей можно дополнить края



Далее если не указано иного предполагается, что якорь свёртки в центре ядра.

### Оператор Собеля

Оператор Собеля является производным оператором, который используется для поиска ребер в изображении, использующий маски производных первого порядка.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \circ A$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \circ A$$

где  $G_x, G_y$  – два изображения, на которых каждая точка содержит приближённые производные по горизонтали и вертикали. Приближённое значение модуля градиента можно вычислить по следующей формуле:

$$G = \sqrt{G_x^2 + G_y^2}$$



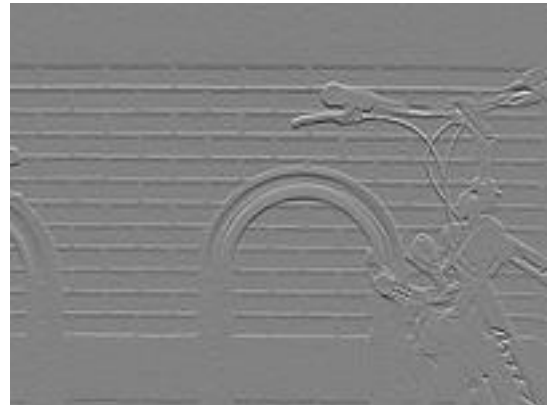
Исходное изображение



Нормализованный градиент изображения



Нормализованный градиент  
изображения по x



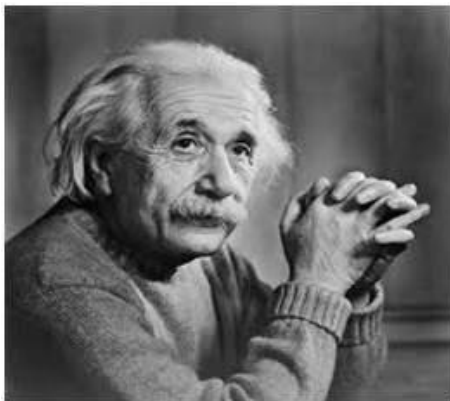
Нормализованный градиент  
изображения по y

### Оператор Лапласа

Оператор Лапласа также является производным оператором, который используется для поиска ребер в изображении, использующий маски второго порядка. В этой маске у нас есть еще две классификации: одна – положительный оператор Лапласа, а другая – отрицательный оператор Лапласа.

Положительный оператор Лапласа используется для удаления внешних краев в изображении:

$$L = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \circ A$$



Исходное изображение



Лапласиан

Отрицательный оператор Лапласа используется для удаления внутренних краев в изображении:

$$L = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \circ A$$

Существует модификация, учитывающая диагональные элементы:

$$L = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \circ A$$

## Размытие по Гауссу

Функция гаусса

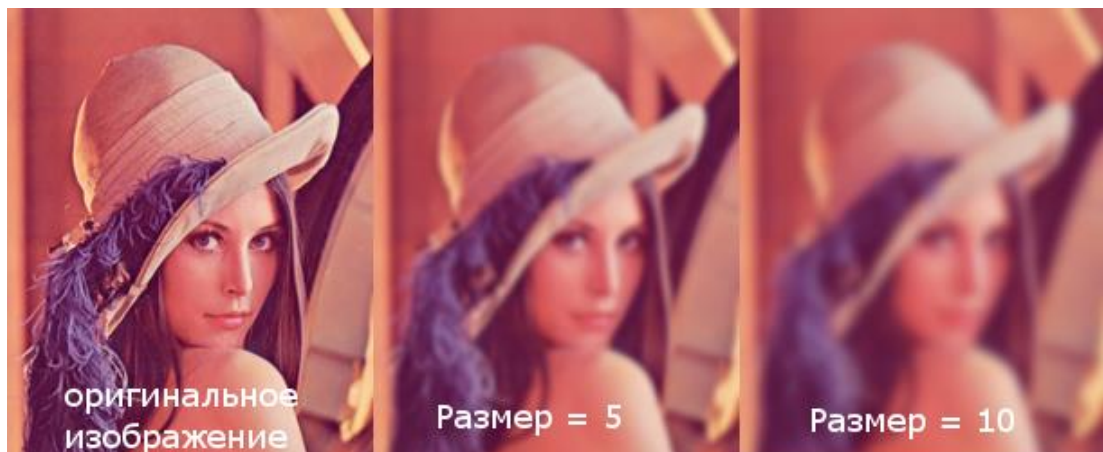
$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

где  $\sigma$  – среднее квадратичное отклонение. В обработке изображений – величина, показывающая силу размытия. Обычно элемент ядра  $F_{h \times w}^{i,j}$ , свёртки высотой  $h$  и шириной  $w$ , определяется следующей формулой:

$$F_{h \times w}^{i,j} = G\left(\frac{h}{2} - i, \frac{w}{2} - j\right)$$

Тогда размытие по гауссу определяется через свёртку следующим способом:

$$G = F_{h \times w} \circ A$$



## Выделение границ

Выделение границ происходит в четыре этапа

- 1) Привидение изображения в градации серого
- 2) Применяется размытие по гауссу
- 3) Вычисляется оператор Лапласа (или Собеля)
- 4) Вычисляется  $G_{ij}$  по заданному порогу

$$G_{ij} = \begin{cases} 1, & L_{ij} > \text{порога} \\ 0, & \text{иначе.} \end{cases}$$

## Медианный фильтр

Для текущего пикселя, пиксели, которые “попадают” в ядро, сортируются, и выбирается среднее значение из отсортированного массива. Это значение и является выходным для текущего пикселя.



original image



1px median filter



3px median filter



10px median filter

## Графический интерфейс

Для создания графического интерфейса разрешено использовать любой язык программирования, однако обработка изображений должна происходить в динамической библиотеке, написанной на языке C.