

Теоретические сведения по обработке изображений для выполнения лабораторной работы №2

1. Получение негатива (для произвольных изображений)

$$R' = 255 - R$$

$$G' = 255 - G$$

$$B' = 255 - B$$

Значения R, G, B – цветовые компоненты исходного изображения.

2. Обработка медианным фильтром (делать только для полутоновых изображений)

Устранение шума с помощью медианного фильтра - выбор медианы среди значений яркости пикселей в некоторой окрестности.

Определение медианы:

$A_i, i = \overline{1, n}$; - отсортированный набор чисел,

$A_{[n/2]}$ – медиана набора.

Медианный фильтр радиусом r – выбор медианы среди пикселей в окрестности $[-r, r]$, т.е. новый цвет пикселя с координатами x, y — это медиана среди всех пикселей в квадрате, ограниченном $x-r, x+r, y-r, y+r$.

3. Построение гистограммы (для полутоновых строится одна гистограмма, для цветных 3 - одна на каждый канал)

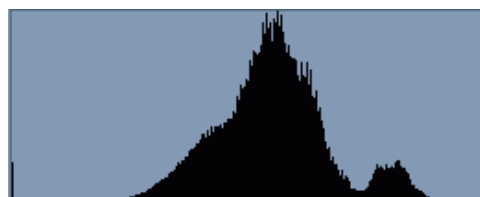
Гистограммы - представления распределения частот выбранных переменных. Каждому интервалу значений переменной соответствует некий счетчик (столбец гистограммы). Значение счетчика (высота столбца) соответствует частоте попадания значения переменной в данный интервал. Гистограмма – это график распределения тонов на изображении. На горизонтальной оси отображается шкала яркостей тонов от белого до черного, а на вертикальной оси указано число пикселей заданной яркости на изображении.

Пример.



0

255



0

255

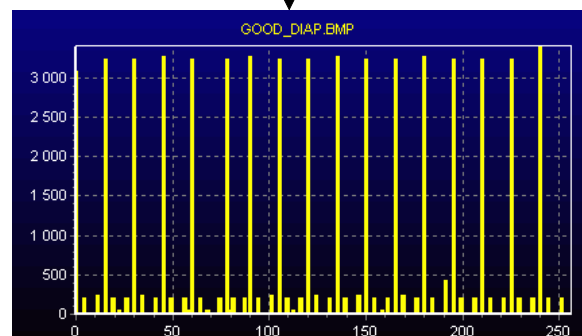
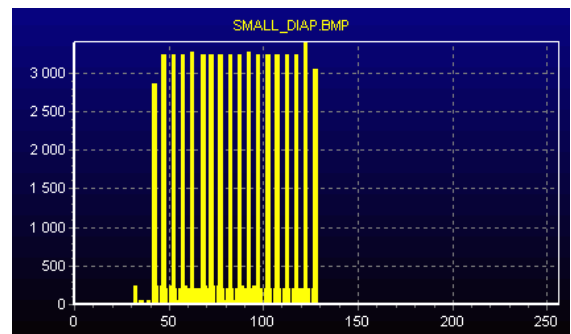
4. Линейная коррекция (для полутоновых применяется к одному каналу, для цветных — к каждому из 3-х по отдельности)

Коррекция - к изображению применяется преобразование яркостей: $y=f(x)$

x – цвет пикселя на исходном изображении,
 y – цвет пикселя после коррекции.

$$y = (x - x_{\min}) * \frac{255}{x_{\max} - x_{\min}}$$

Компенсация узкого диапазона яркостей – линейное растяжение:

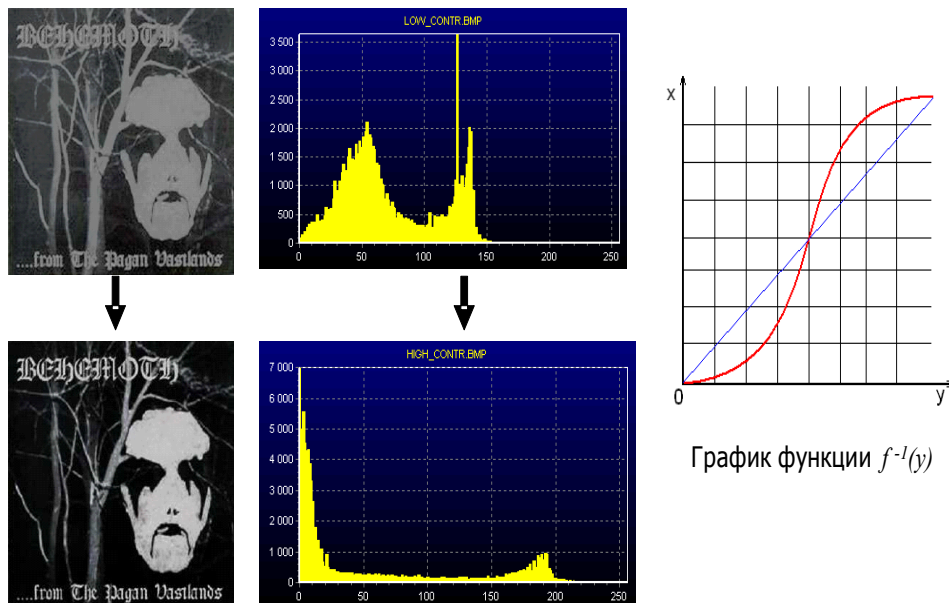


5. Нелинейная коррекция (для полутоновых применяется к одному каналу, для цветных — к каждому из 3-х по отдельности)

Нелинейная компенсация недостаточной контрастности (например, логарифмическая)
Цель – сжатие динамического диапазона при визуализации данных.

$$y = c \cdot \log(1 + x)$$

Где c — некоторая константа (задается пользователем)



6. Выравнивание яркости (делать только для полутоновых изображений)

Задача заключается в том, чтобы исправить неравномерную освещенность (засветку, блики и т.д.). Решение отыскивается в несколько этапов:

- 1) получение изображения, отфильтрованного фильтром (например, Гаусса)

$$I'(i, j) = \sum_{l=-n}^n \sum_{k=-m}^m I(i-l)(j-k) \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{d^2}{2\sigma^2}}$$

$$d = \sqrt{l^2 + k^2}$$

параметр σ задает степень размытия (задает пользователь в диапазоне 1-10)

- 2) расчет новых параметров пикселя

$$new_color = old_color * max / (gauss_color + 1),$$

где

new_color – новый цвет

old_color – старый цвет

gauss_color – цвет после фильтрации

max – максимальное значение цвета в изображении

- 3) нормализация цветов в диапазон 0-255

6. Поворот изображения (для произвольных изображений)

Новые координаты каждого пикселя рассчитываются по формулам:

$$x(k, l) = (k, x0)\cos(\mu) + (l, y0)\sin(\mu) + x0;$$

$$y(k, l) = (k, x0)\sin(\mu) + (l, y0)\cos(\mu) + y0;$$

x_0, y_0 - центр поворота (задается пользователем),

μ - угол поворота (задается пользователем).

Если пиксель попадает за пределы изображения, он отбрасывается. Пиксели, которые становятся свободными — им присваивается значение фона (задается пользователем)

7. Фильтрация произвольным фильтром (для произвольных изображений)

Фильтр всегда задается некоторой матрицей (чаще всего, квадратной $n \times n$, где n - нечетное).

$$F = \begin{bmatrix} F_{1,1} & \dots & F_{1,n} \\ \dots & \dots & \dots \\ F_{n,1} & \dots & F_{n,n} \end{bmatrix}$$

В этом случае значение цвета пикселя вычисляется по формуле :

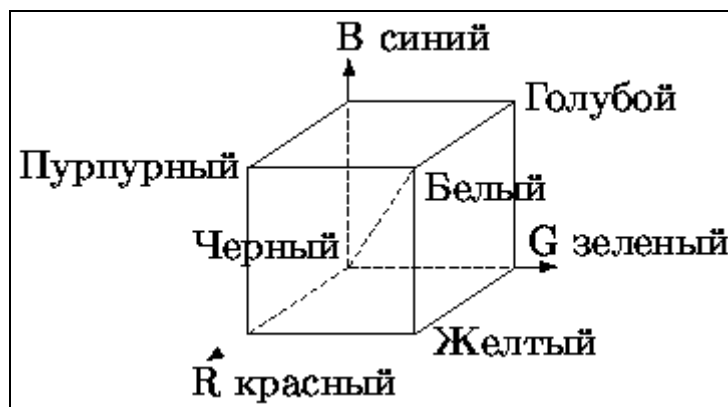
$$I(x, y) = \frac{1}{n^2} \sum_{i=-n/2}^{n/2} \sum_{j=-n/2}^{n/2} I(x-i, y-j) * F(i+n/2, j+n/2)$$

Размерность фильтра и значения коэффициентов задаются пользователем.

8. Преобразование цветовых моделей (для произвольных изображений)

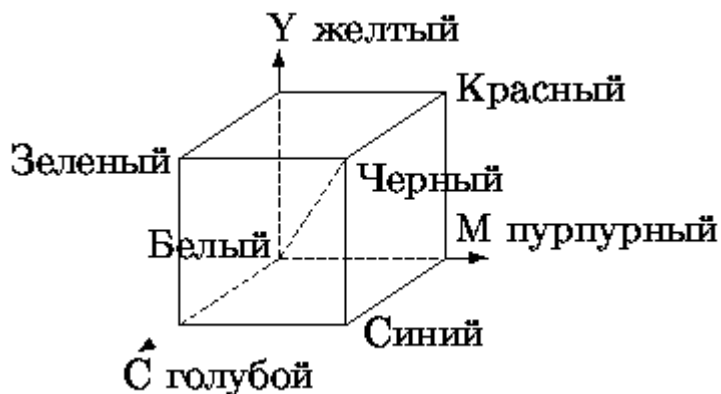
Назначение цветовой модели - дать средства описания цвета в пределах некоторого цветового охвата, в том числе и для выполнения интерполяции цветов. Наиболее часто в компьютерной графике используются модели RGB, CMY, YIQ, HSV и HLS.

RGB (Red, Green, Blue - красный, зеленый, синий) - аппаратно-ориентированная модель, используемая в дисплеях для аддитивного формирования оттенков самосветящихся объектов (пикселей экрана). Система координат RGB - куб с началом отсчета (0,0,0), соответствующим черному цвету.. Максимальное значение RGB - (255,255,255) соответствует белому цвету.



Модель RGB

CMY (Cyan, Magenta, Yellow - голубой, пурпурный, желтый) - аппаратно-ориентированная модель, используемая в полиграфии для субтрактивного формирования оттенков, основанного на вычитании слоев краски части падающего светового потока. Цвета модели CMY являются дополнительными к цветам модели RGB, т.е. дополняющими их до белого. Таким образом система координат CMY - тот же куб, что и для RGB, но с началом отсчета в точке с RGB координатами (255,255,255), соответствующей белому цвету.



Цветовой куб модели CMY

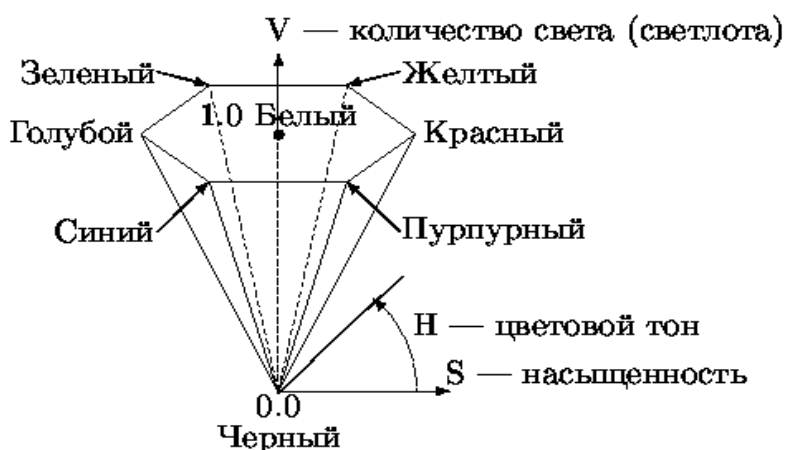
Преобразования между пространствами RGB и CMY определяются следующим образом: $[R\ G\ B] = [255\ 255\ 255] - [C\ M\ Y]$.

Причем единичный вектор-строка в модели RGB - представление белого цвета, а в модели CMY - черного.

HSV

HSV (Hue, Saturation, Value - цветовой тон, насыщенность, количество света или светлота) - модель, ориентированная на человека и обеспечивающая возможность явного задания требуемого оттенка цвета.

Подпространство, определяемое данной моделью - перевернутый шестигранный конус.



По вертикальной оси конуса задается V - светлота, меняющаяся от 0 до 255. Значению $V = 0$ соответствует вершина конуса, значению $V = 255$ - основание конуса; цвета при этом наиболее интенсивны.

Цветовой тон H задается углом, отсчитываемым вокруг вертикальной оси. В частности, 0° - красный, 60° - желтый, 120° - зеленый, 180° - голубой, 240° - синий, 300° - пурпурный, т.е. дополнительные цвета расположены друг против друга (отличаются на 180°).

Насыщенность S определяет насколько близок цвет к "чистому" пигменту и меняется от 0 на вертикальной оси V до 255 на боковых гранях шестигранного конуса.

Точка V = 0, в которой находится вершина конуса, соответствует черному цвету. Значение S при этом может быть любым в диапазоне 0-255. Точка с координатами V = 255, S = 0 - центр основания конуса соответствует белому цвету. Промежуточные значения координаты V при S=0, т.е. на оси конуса, соответствуют серым цветам. Если S = 0, то значение оттенка H считается неопределенным.

```
//На входе R,G,B - цвет пикселя, на выходе H,S,V - цвет
//пикселя в другой системе
begin
    max_value:=max_of(R,G,B);
    min_value:=min_of(R,G,B);
    diff:=max_value-min_value;
    V:=max_value;
    if max_value<>0 then s:=(diff / max_value) else S:=0;
    if S=0 then h:=undefined
        else
            begin
                r:=(max_value-R) / diff;
                g:=(max_value-G) / diff;
                b:=(max_value-B) / diff;
                if R=max_value
                    then H:=b_dist-g_dist
                    else if G=Max_value
                        then H:=2+r_dist-b_dist
                        else if B=max_value
                            then H:=4+g_dist-r_dist;
                H:=H*60;
                if H<0 then H:=H+360;
            end;
end;
```

9. Градиентный метод выделения контуров (реализовывать только для полутоновых изображений)

Контур (граница) на изображении — это множество пикселей, в которых наблюдается резкое изменение (перепад) яркости или цвета. Самый простой способ — оценивать градиент (изменение) по величине и направлению.

$$|\nabla g(x, y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2}$$

Для того, чтобы не вычислять производную, переходят к конечным разностям и формула для градиента приобретает вид:

$$|\nabla g(x, y)| = \sqrt{[g(x+1, y) - g(x, y)]^2 + [g(x, y+1) - g(x, y)]^2}$$

Алгоритм поиска границы заключается в том, что для каждого пикселя оценивается

значение градиента и если модуль градиента больше некоторого порога T , то пиксель считается лежащим на границе. Такие пиксели необходимо пометить каким-либо цветом (например, белым), а все остальные либо оставить без изменения, либо заменить на цвет фона. Выбор варианта отображения результата и величины порога T (1-255) осуществляется пользователем.

10. Метод наращивания участков (реализовывать только для полутоновых изображений)

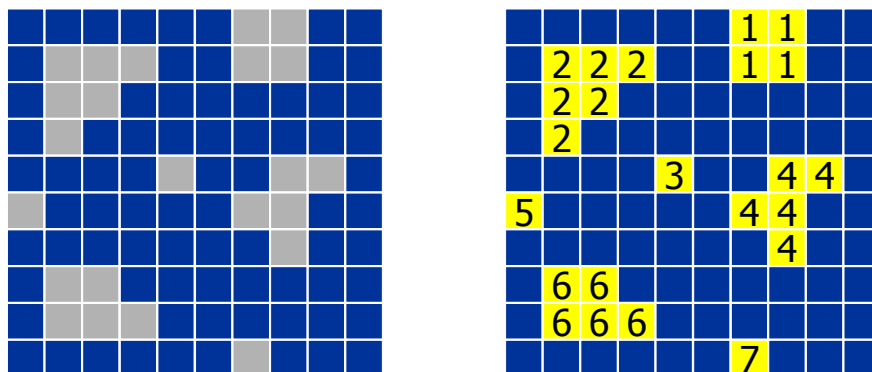
Используем итерационный алгоритм для разбиения по яркости, так называемый алгоритм роста регионов (Region Growing). Идея — пометить, как одинаковые на изображении связанные области, обладающие некоторой степенью сходства по цвету (яркости).

Начинаем обход, для определённости, из левого верхнего угла.

1. Объявляем левый верхний пиксель изображения новым классом.
2. Для пикселей первой строки считаем отклонение от класса левого пикселя и, сравнивая с порогом, либо добавляем пиксель к классу соседа, либо заводим новый класс.
3. Первый пиксель каждой последующей строки сравниваем с классом верхнего пикселя. Далее будем сравнивать пиксель с классами двух соседей: левого и верхнего.
 1. Если отклонение от обоих классов больше порога, то заводим новый класс, если отклонение больше только для одного класса, то добавляем пиксель к тому классу отклонение от которого меньше порога.
 2. Если отклонение допустимо для обоих классов, то возможно 2 варианта:
 1. $L(g(C_1) - g(C_2)) < \delta$ - тогда объединяем эти 2 класса (если они не один и тот же класс) и добавляем к объединённому классу текущий пиксель.
 2. $L(g(C_1) - g(C_2)) > \delta$ - в этом случае добавляем пиксель к тому классу, от которого отклонение минимально.

В алгоритме под “отклонением” понимается модуль разности между яркостями пикселей. Порог и параметр σ задаются пользователем.

Пример работы алгоритма:



11. Зашумление (для произвольных изображений)

К каждому цветовому компоненту пикселя добавляется некоторая поправка (шум), которая генерируется случайным образом.

Например:

$$C' = C + (Noise * 255 - 127)$$

где

Noise - уровень шума (0...1)

C - старый цвет

C' - новый цвет

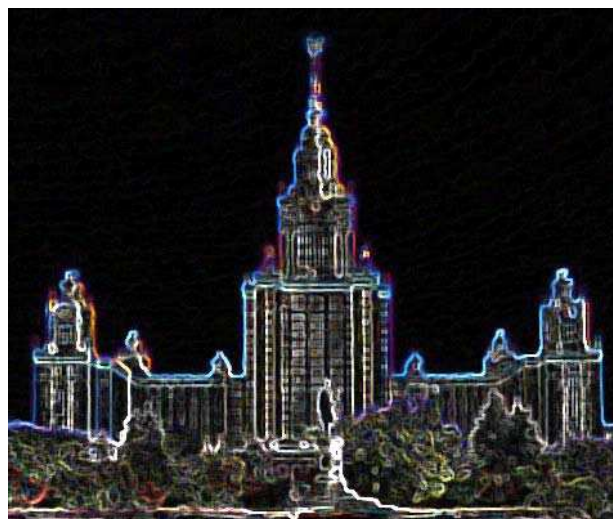
При расчете формулы следует учитывать, что значение может стать меньше 0 или больше 255. В этом случае необходимо принудительно ограничивать значение цвета.

Уровень шума вводится пользователем.

12. Эффект «горящих краев»

Достигается последовательным применением таких этапов:

- 1) фильтрация медианным фильтром (п.2)
- 2) выделение границ градиентным методом (п. 9)
- 3) применение фильтра максимума (т.е. поиск тех пикселей, для которых значение градиента максимально и их выделение некоторым цветом)

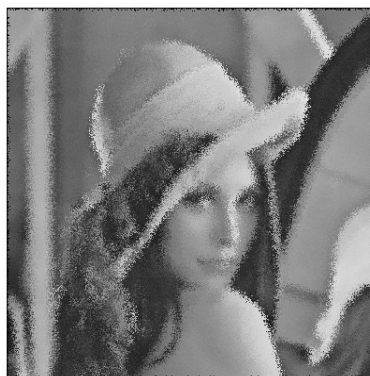


13. Эффект «стекла» (для произвольных изображений)

Расчет новых координат пикселей по формулам:

$$x(k, l) = k + (\text{rand}(1, 1) - 0.5) * 10;$$

$$y(k, l) = l + (\text{rand}(1, 1) - 0.5) * 10;$$



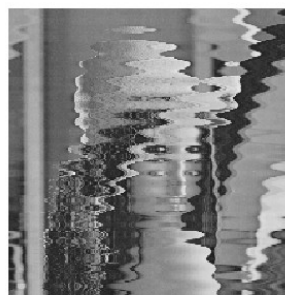
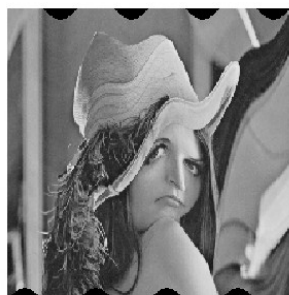
14. Эффект «волны» (для произвольных изображений)

Расчет новых координат пикселей по формуле:

$$x(k, l) = k + 20\sin(2\pi l / 128); y(k, l) = l;$$

или

$$x(k, l) = k + 20\sin(2\pi k / 30); y(k, l) = l;$$



15. Сегментация методом «разделения» (реализовывать для полутоновых изображений)

1. Первый шаг — всё изображение это одна область, поместить область в стек

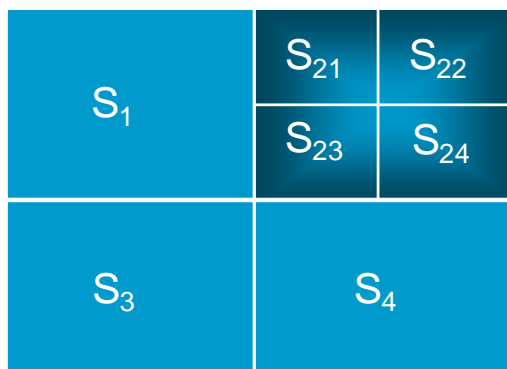
2. Пока стек не пуст

- Взять область S из стека
- Проверить область на однородность
- Если область неоднородна
 - разделить ее, новые области поместить в стек
- Если область однородна
 - область больше не трогаем

Область — это множество пикселей

Однородность области означает, что значение цветов всех пикселей в ней отличаются друг от друга не более, чем на некоторый порог T .

Правило деления областей — на 4 части, как квадродерево



Пример работы алгоритма:

1	1	1	1	1	1	1	2
1	1	1	1	1	1	1	0
3	1	4	9	9	8	1	0
1	1	8	8	8	4	1	0
1	1	6	6	6	3	1	0
1	1	5	6	6	3	1	0
1	1	5	6	6	2	1	0
1	1	1	1	1	1	0	0

16. Сегментация методом «слияния» (реализовывать только для полутоновых изображений)

1.Первый шаг – каждый пиксель это отдельная область, поместить все области в стек

2.Пока стек не пуст

- Взять область S из стека, для всех соседних областей S_i :

- Проверить $S' = S \cup S_i$ на однородность

- Если S' однородна -

- Слить S и S_i , S' поместить в стек, S_i из стека удалить, перейти на 2

- Если область не однородна

- Пробуем другого соседа

Область — это множество пикселей

Однородность области означает, что значение цветов всех пикселей в ней отличаются друг от друга не более, чем на некоторый порог T .

17. Кластеризация (реализовывать только для полутоновых изображений)

Способ определения нескольких порогов одновременно

Нужно заранее знать k - количество диапазонов яркостей (можно вводить пользователю)

Входные данные – набор пикселей изображения v_i $i=1, \dots, p$.

Выходные данные – центры кластеров m_j $j=1, \dots, k$ и принадлежность v_i к кластерам

1.Случайным образом выбрать k средних m_j $j=1, \dots, k$;

2.Для каждого v_i $i=1, \dots, p$ подсчитать расстояние до каждого из m_j $j=1, \dots, k$,

3.Отнести (приписать) v_i к кластеру j' , если расстояние до $m_{j'}$ минимально;

4.Пересчитать средние m_j $j=1, \dots, k$ по всем кластерам;

5.Повторять шаги 2, 3 пока кластеры не перестанут изменяться;

18. Коррекция гаммы (для произвольного изображения)

Коррекция цвета пикселя в соответствии с формулой

$$y = c * x^{\gamma}$$

Параметры c и γ вводятся пользователем

19. Фильтр Sobel (для произвольного изображения)

Фильтрация фильтром в соответствии с п. 7, используя матрицу

$$h_1 = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

20. Фильтр Canny (реализовывать только для полутоновых изображений)

Фильтрация фильтром Гаусса в соответствии с п. 6.1.

Далее для каждого пикселя рассчитывается градиент (п. 9) и направление по формуле

$$direct = \arctg\left(\frac{\nabla y}{\nabla x}\right)$$

Направляющий угол округляется до одного из базовых (0, 45, 90, 135).

Далее применяются следующие правила. Пиксель считается лежащим на границе, если:

- угол равен 0 и значение яркости больше, чем яркость по направлению вверх и вниз;
- угол равен 90 и значение яркости больше, чем яркость по направлению влево и вправо;
- угол равен 45 и значение яркости больше, чем яркость по направлению вниз-вправо и вверх-влево;
- угол равен 135 и значение яркости больше, чем яркость по направлению вниз-влево и вверх-вправо.

21. Фильтр Prewitt (для произвольных изображений)

Фильтрация фильтром в соответствии с п. 7, используя матрицу

$$h_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$$

22. Преобразование RGB -> Y'CbCr (JPEG)

Формат обмена файлами JPEG позволяет использовать Y'CbCr, где Y, C_B и C_R имеют полный 8-битный диапазон 0-255:

$$\begin{aligned} Y' &= \quad + (0.299 \cdot R'_D) + (0.587 \cdot G'_D) + (0.114 \cdot B'_D) \\ C_B &= 128 - (0.168736 \cdot R'_D) - (0.331264 \cdot G'_D) + (0.5 \cdot B'_D) \\ C_R &= 128 + (0.5 \cdot R'_D) - (0.418688 \cdot G'_D) - (0.081312 \cdot B'_D) \end{aligned}$$

И обратно:

$$\begin{aligned} R &= Y + 1.402 \cdot (C_R - 128) \\ G &= Y - 0.34414 \cdot (C_B - 128) - 0.71414 \cdot (C_R - 128) \\ B &= Y + 1.772 \cdot (C_B - 128) \end{aligned}$$

23. Простая конверсия

$$\begin{aligned} R &= Y + 1.13983 \cdot V; \\ G &= Y - 0.39465 \cdot U - 0.58060 \cdot V; \\ B &= Y + 2.03211 \cdot U; \end{aligned}$$

$$\begin{aligned} Y &= 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B; \\ U &= -0.14713 \cdot R - 0.28886 \cdot G + 0.436 \cdot B; \\ V &= 0.615 \cdot R - 0.51499 \cdot G - 0.10001 \cdot B; \end{aligned}$$