

Обработка изображений - 2

ЛКШ.2017.Зима.С'

Шумоподавление

Причины возникновения шума:

- Несовершенство измерительных приборов
- Хранение и передача изображений с потерей данных



Шум фотоаппарата



Сильное сжатие JPEG

Медианная фильтрация



Задание 13: Медианная фильтрация

Задача: избавиться от шума типа соль и перец - очень яркие или очень тёмные отдельные пиксели

Идея: для каждого пикселя посмотрим на соседние пиксели $(x-1..x+1, y-1..y+1)$, отсортируем их в порядке увеличения яркости и возьмём средний в качестве нового цвета пикселя

Что будет, если увеличить размер просматриваемой области?



Линейные фильтры

0	0.2	0
0.2	0.2	0.2
0	0.2	0

1	5	10	3	1	9	4	7	9	2
1	1	2	6	4	6	9	0	0	7
3	4	4	7	0	9	1	1	2	1
6	9	0	8	5	5	5	1	3	3
6	3	3	2	2	3	4	7	1	8
7	3	0	5	1	9	8	2	1	4
6	7	9	0	3	3	3	8	8	8
8	4	4	4	4	5	7	1	1	1
2	9	9	9	0	0	0	4	4	8
2	9	0	0	0	4	4	4	7	7

5

$$\begin{aligned} 5 = & \\ & 0 * 2 + 0.2 * 6 + 0 * 4 + \\ & 0.2 * 4 + 0.2 * 7 + 0.2 * 0 + \\ & 0 * 0 + 0.2 * 8 + 0 * 5 \end{aligned}$$

Задание 14: Размытие (Box filter)

Задача: избавиться от нормального шума

Идея: для каждого пикселя в качестве нового среднего возьмем среднее арифметическое соседних пиксели ($x-1..x+1$, $y-1..y+1$).

Что будет, если усреднять по большему количеству соседних пикселей, например ($x-9..x+9$, $y-9..y+9$)?

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



Задание 15: Любой фильтр

Задача: написать функцию для применения фильтра размера $kw * kh$. В фильтре могут быть любые значения.

Параметры: kernel - ядро фильтра, задаётся константой в коде

Как:

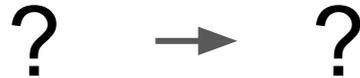
- нормализовать фильтр: разделить каждый компонент на сумму значений по всему фильтру
- посчитать значения для каждого пикселя в новом изображении

Проблема:

На границе изображения ядро выходит за границу

Возможные решения:

- Создавать изображение с шириной и высотой меньше
- Считать цветом пикселя с координатами, выходящими за границу изображения, цвет ближайшего пикселя изображения



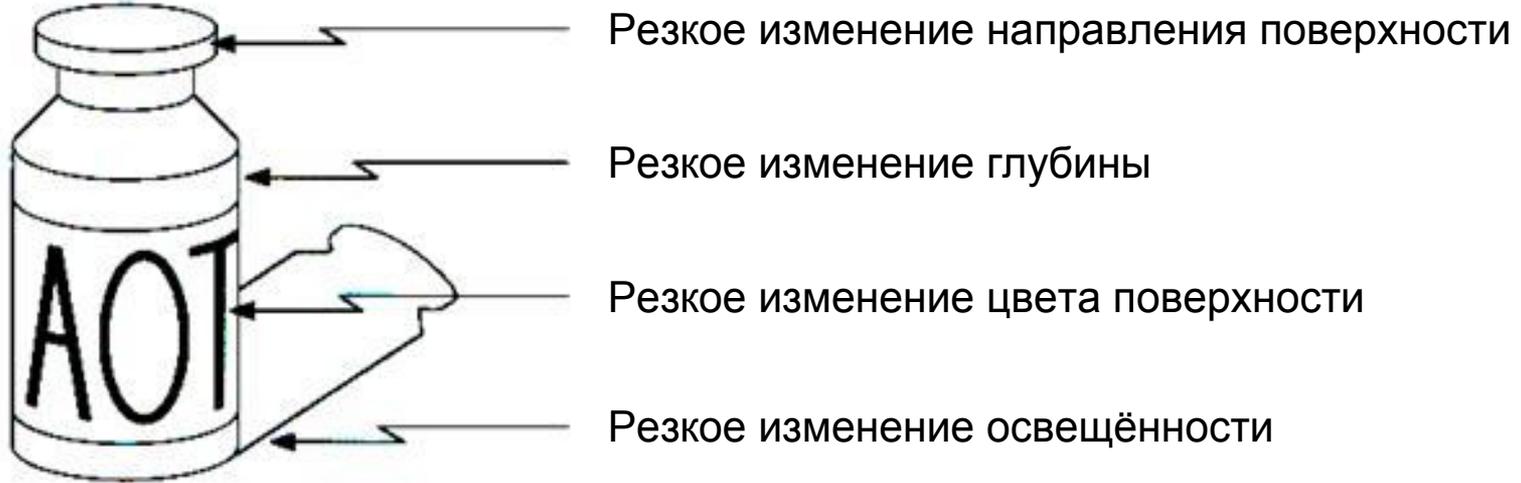
Выделение краёв

Задача: Выделить резкие изменения (разрывы) изображения

Информация о краях активно используется при распознавании изображений



Откуда берутся края



Край - точка резкого изменения яркости изображения

Задание 16: Выделение краёв

Задача: найти края на изображении

Идея: подсчитать изменение цвета пикселей по осям x и y и найти их среднее квадратичное

Как:

- $r_dx = R(x + 1, y) - R(x, y)$
 $r_dy = R(x, y + 1) - R(x, y)$
 $R'(x, y) = \sqrt{(r_dx ** 2 + r_dy ** 2) / 2}$
- Аналогично для остальных компонент



Проблема: на границе изображения $x + 1$ и $y + 1$ выходят за границу

Задание 17: Фильтр Собеля

Задача: найти края на изображении фильтром Собеля

Идея: подсчитать изменение цвета пикселей по осям x и y и найти их среднее квадратичное

Как:

- $r_dx = \text{apply_kernel}(\text{first}, x, y)[R]$
 $r_dy = \text{apply_kernel}(\text{second}, x, y)[R]$
 $R'(x, y) = \text{sqrt}((r_dx ** 2 + r_dy ** 2) / 2)$
- Аналогично для остальных компонент

first =

-1	0	1
-2	0	2
-1	0	1

second =

1	2	1
0	0	0
-1	-2	-1



Стандартные фильтры PIL

```
from PIL import ImageFilter
```

```
img1 = img.filter(ImageFilter.BLUR)
```

Другие фильтры:

```
ImageFilter.MedianFilter
```

```
ImageFilter.EDGE_ENHANCE
```

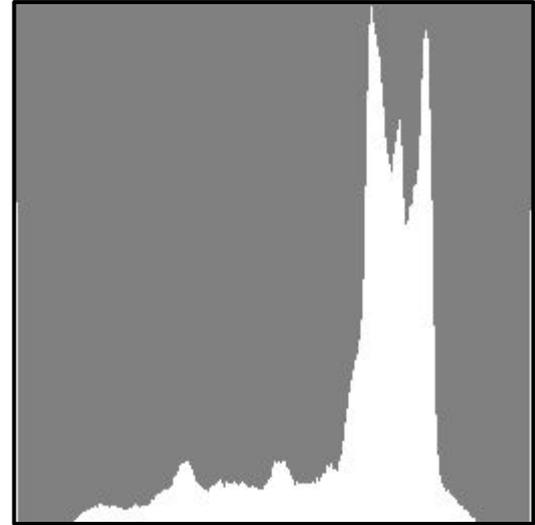
Свой фильтр:

```
ImageFilter.Kernel(size, kernel)
```

Задание 18: Гистограмма

Задача: Построить гистограмму яркости заданного изображения.

Идея: Сгенерировать изображение 256x256. Высота столбца с номером i зависит от количества пикселей с яркостью i . Полностью заполненный столбец соответствует максимальному количеству пикселей с заданной яркостью.



Старые слайды

Виды шума



Original



Salt and pepper noise

Соль и перец: случайные чёрные и белые пиксели

Импульсный: случайные белые пиксели

Нормальный: случайные отклонения яркости



Impulse noise



Gaussian noise