

# Основные операции с изображениями

## Фильтры и их применение

Диц Даниил Денисович

НТУ Сириус

7 марта 2025 г.

# Операции с изображениями

- ▶ Логические
- ▶ Статистические
- ▶ Геометрические
- ▶ Математические
- ▶ Аффинные преобразования

# Распределение интенсивности пикселей

Математическое представление:

$$H(k) = \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \delta(I(i,j) - k)$$

где:

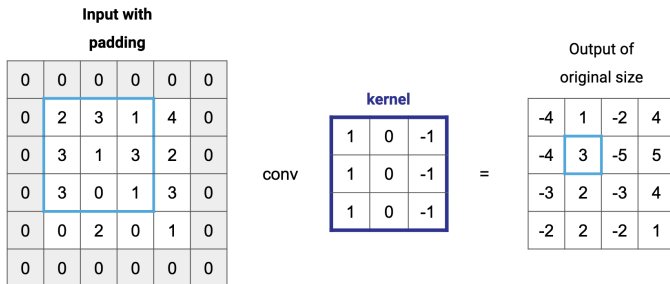
- ▶  $H(k)$  - количество пикселей с интенсивностью  $k$
- ▶  $I(i,j)$  - интенсивность в точке  $(i,j)$
- ▶  $\delta$  - функция Кронекера

## Важность

Анализ гистограммы помогает:

- ▶ Оценить контраст изображения
- ▶ Выявить переэкспозицию
- ▶ Подобрать параметры обработки

# Операция свертки



# Операция свертки

Основная формула:

$$(f * g)(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k f(x-i, y-j) \cdot g(i, j)$$

Где:

- ▶  $f$  - исходное изображение
- ▶  $g$  - ядро свертки (фильтр)
- ▶  $k$  - радиус ядра

## Примеры ядер

Размытие:

$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Собель (X):

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

```
blur = cv.GaussianBlur(img, (5,5), 0)
```

```
sobel_x = cv.Sobel(img, cv.CV_64F, 1, 0, ksize=5)
```

### Важные аспекты

- ▶ Размер ядра влияет на силу эффекта
- ▶ Граничные условия (padding) важны для обработки краев
- ▶ Нормализация значений сохраняет диапазон интенсивностей

# Аффинный сдвиг

Матрица преобразования:

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

Формула преобразования:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

```
def translate(img, x, y):  
    transMat = np.float32(  
        [[1, 0, x],  
         [0, 1, y]])  
    return cv.warpAffine(  
        img, transMat,  
        (img.shape[1],  
         img.shape[0]))
```

# Поворот

Матрица поворота:

$$M = \begin{bmatrix} \alpha & \beta & (1 - \alpha)x_0 - \beta y_0 \\ -\beta & \alpha & \beta x_0 + (1 - \alpha)y_0 \end{bmatrix}$$

где:

- ▶  $\alpha = \cos \theta$
- ▶  $\beta = \sin \theta$
- ▶  $(x_0, y_0)$  - центр вращения



# Гауссово размытие (GaussianBlur)

Функция ядра Гаусса:

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

Операция свертки:

$$I'(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x+i, y+j) \cdot G(i, j)$$

# Детектор границ Канни (Canny)

Основные этапы:

1. Вычисление градиентов:

$$G_x = \frac{\partial I}{\partial x}, \quad G_y = \frac{\partial I}{\partial y}$$

2. Магнитуда и направление:

$$|G| = \sqrt{G_x^2 + G_y^2}, \quad \theta = \arctan\left(\frac{G_y}{G_x}\right)$$

3. Подавление немаксимумов
4. Гистерезисная фильтрация (125, 175)

# Поиск особенностей

## Определение

Особенности (features) — это уникальные точки или области на изображении, которые можно выделить и описать.

## Примеры особенностей

- ▶ Углы (например, углы зданий)
- ▶ Границы (например, края объектов)
- ▶ Текстуры (например, узоры на ткани)

## Требования к особенностям

- ▶ Устойчивость к изменениям масштаба
- ▶ Устойчивость к поворотам
- ▶ Устойчивость к изменениям освещения

# Построение SIFT дескриптора

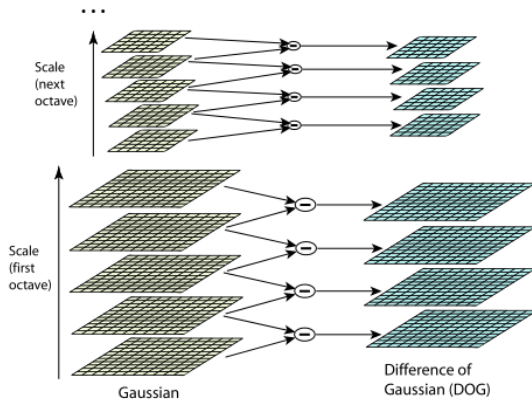


Рис.: разница Гауссианов

# Построение SIFT дескриптора

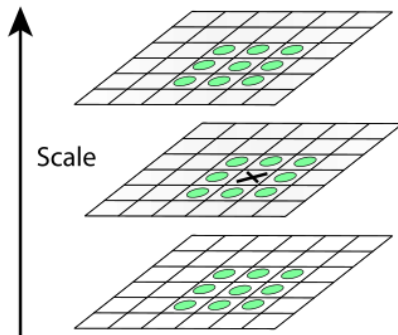


Рис.: Поиск сквозного экстремума

# Основные этапы сопоставления

## 1. Детекция ключевых точек

- ▶ Алгоритмы: ORB, SIFT, SURF
- ▶ ORB: быстрый, работает в реальном времени
- ▶ SIFT: более точный, но запатентован

## 2. Вычисление дескрипторов

- ▶ Дескриптор — вектор, описывающий область вокруг ключевой точки
- ▶ ORB: бинарные дескрипторы
- ▶ SIFT: вещественные дескрипторы

## 3. Сопоставление дескрипторов

- ▶ Brute-Force: попарное сравнение всех дескрипторов
- ▶ FLANN: быстрый, но приближенный метод

# Применение сопоставления особенностей

## Поиск объектов

- ▶ Сравнение особенностей эталонного изображения с целевым
- ▶ Пример: поиск лица на фотографии

## Сшивка изображений

- ▶ Поиск общих точек для объединения снимков
- ▶ Пример: панорамные фотографии

## 3D-реконструкция

- ▶ Восстановление 3D-структуры по нескольким 2D-изображениям
- ▶ Пример: создание 3D-моделей зданий

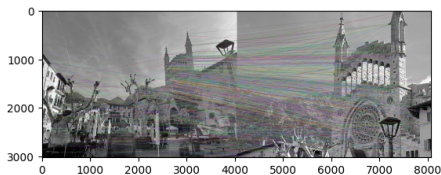


Рис.: Пример сопоставления особенностей