

Лабораторная работа №6

Преобразование Хафа

Цель работы: изучить принцип применения преобразования Хафа для поиска прямых и окружностей

Основная информация

Стандартное преобразование Хафа для поиска прямых:

```
lines = cv2.HoughLines(  
    img_bin, #входное изображение  
    rho_res, #разрешение по расстоянию  
    theta_res, #разрешение по углу  
    threshold, #значение аккумулятора  
    min_theta=None, #ограничение угла  
    max_theta=None #ограничение угла  
)
```

Возвращаемое значение - список коэффициентов для всех обнаруженных прямых.

Структура lines:

Для line в lines:
(rho, theta) = line[0]

Многомасштабное преобразование Хафа для поиска прямых:

```
lines = cv2.HoughLines(  
    img_bin, #входное изображение  
    rho_res, #разрешение по расстоянию  
    theta_res, #разрешение по углу
```

```
threshold, #значение аккумулятора  
min_theta=None, #ограничение угла  
max_theta=None, #ограничение угла  
srn=srn, #уточнение расстояния  
stn=stn #уточнение угла  
)
```

Уточнение расстояния и уточнение угла позволяют увеличить разрешение по соответствующей величине:

```
rho = rho_res / srn  
theta = theta_res / stn
```

Структура lines - та же, что и для стандартного преобразования Хафа

Прогрессивное вероятностное преобразование Хафа для поиска прямых:

```
linesP = cv2.HoughLinesP(  
    img_bin,  
    rho_res,  
    theta_res,  
    threshold,  
    minLineLength=length,  
    maxLineGap=gap  
)
```

Преобразование позволяет искать отрезки, а не прямые

minLineLength – минимальная длина отрезка

maxLineGap – максимальное расстояние между пикселями на одной прямой

Возвращаемое значение - список элементов, которые хранят координаты начала и координаты конца отрезков.

Для line в lines:

```
(x0, y0, x1, y1) = line[0]
```

Преобразование Хафа для поиска окружностей:

```
circles = cv2.HoughCircles(  
    img,  
    cv2.HOUGH_GRADIENT, # метод - всегда  
    dp,  
    minDist, # расстояние между  
окружностями  
    CannyTh, # верхняя граница  
    accTh, # порог аккумулятора  
    minRadius=0,  
    maxRadius=0  
)
```

dp – во сколько раз меньше разрешение у аккумулятора, чем у входного изображения

Возвращаемое значение - список элементов, каждый из которых содержит координаты центра окружности и ее радиус.

Для c в circles:

```
(x0, y0, r) = c[0]
```

Для рисования линий можно воспользоваться функцией line() в OpenCV, для рисования окружностей - функцией circle() в OpenCV. **Подсказки по использованию есть в лекции!**

Задание

При оформлении кода следует придерживаться стандарта PEP 8. Здесь дано его описание на русском языке: <https://pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html>

Основные задания:

Задание 1. Для выполнения этого задания нужно использовать файл 6_1.png. На этом изображении выделите цветом самую большую окружность и самый длинный отрезок. Если сразу не получается, рекомендую самостоятельно подготовить чуть более простое изображение и потренироваться на нем. Начните с изображения, содержащего один отрезок, два, одну окружность и так далее.

Задание 2. Для выполнения этого задания нужно использовать файл 6_2.png. Исправьте это изображение так, чтобы линии таблицы исчезли, а числа остались. Для решения этой задачи воспользуйтесь преобразованием Хафа для поиска прямых.

Отчет

Отчет должен содержать:

- 1) Титульный лист
- 2) Задания
- 3) Исходный код, отформатированный по стандарту PEP 8, с подсветкой синтаксиса
- 4) Комментарии к алгоритму, выбору флагов и т.д.
- 5) Необходимые изображения, иллюстрирующие правильность выполнения задания