

# Лабораторная работа №12. Выделение границ объектов на полутоновом изображении.

Грибчук Даниил

29 сентября 2020 г.

## Описание алгоритма

Для выделения границ на полутоновом изображении я использовал алгоритм выделения границ Канни. Алгоритм состоит из следующих этапов:

1. **Сглаживание.** Размытие изображения для удаления шума.

Для размытия изображения использовался фильтр Гаусса. На практике лучше всего оказался фильтр Гаусса порядка 7 и  $\sigma = 1.0$ .

2. **Поиск градиентов.** Границы отмечаются там, где градиент изображения приобретает максимальное значение.

Для поиска градиентов использовался оператор Собеля. На выходе получаем матрицу градиентов и матрицу направления яркости.

3. **Подавление не-максимумов.** Только локальные максимумы отмечаются как границы.

Рассматриваем пиксель, если его значение градиента больше чем соседних по направлению, то яркость этого пикселя нормированный градиент, иначе 0.

4. **Двойная пороговая фильтрация.** Потенциальные границы определяются порогами.

Выбираем значение верхней и нижней границы. Если значение пикселя выше верхней границы – он принимает значение *strong*, если ниже – пиксель принимает *weak*, точки со значением, попадающим в диапазон между порогов, принимают фиксированное значение *medium*.

Константы этого пункта оказывают ощутимый результат на выделение границ. Чем меньше порог, тем больше границ будет находиться, но тем более восприимчивым к шуму станет результат, выделяя лишние данные изображения. Наоборот, высокий порог может проигнорировать слабые края или получить границу фрагментами.

На практике лучше всего оказались значения 5% и 10% яркости для нижнего и верхнего порога соответственно.

5. **Трассировка области неоднозначности.** Итоговые границы определяются путём подавления всех краёв, несвязанных с определенными (сильными) границами.

Рассматриваем *medium* пиксели, если хотя бы один из 8 соседей является *strong* пикселем, то рассматриваемый пиксель становится *strong*.

Я использовал упрощенную реализацию и из-за этого страдает яркость пикселей на границе. Чтобы это исправить необходимо расширить матрицу изображения, после завершения обработки обрезать ее.

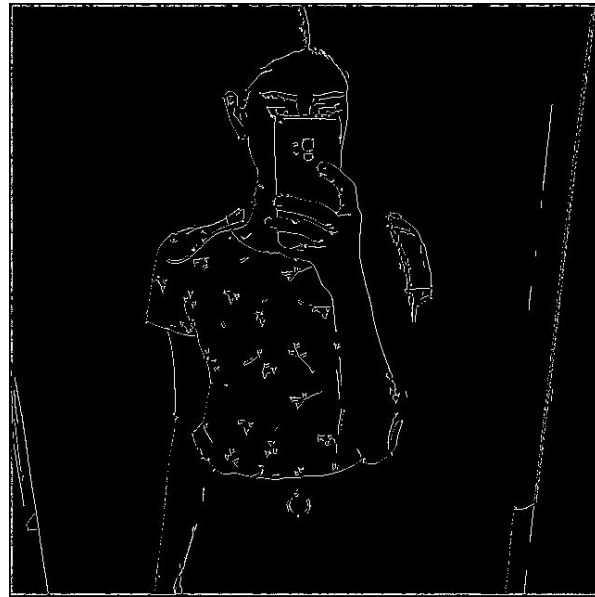
## Ссылка на реализацию

### Пример запуска:

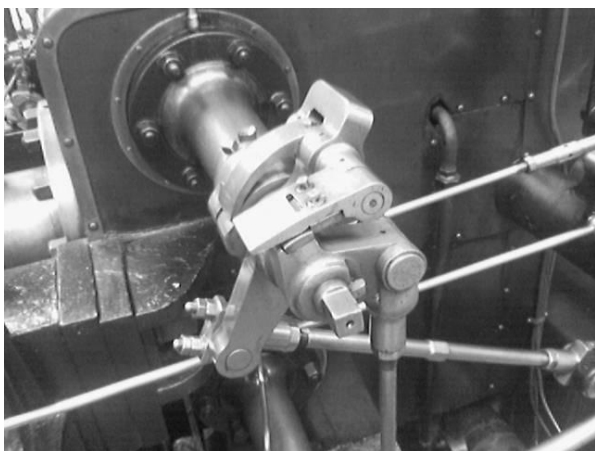
```
python3 canny_edge_detector.py -name="grayscale.jpg" -size=7 -sigma=1.0 -low=0.05 -upp=0.1 -path="border.jpg"
```



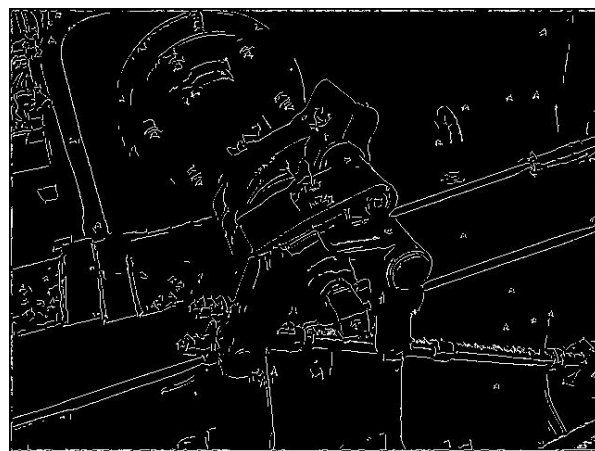
Изображение в полутонах



Выделение границ



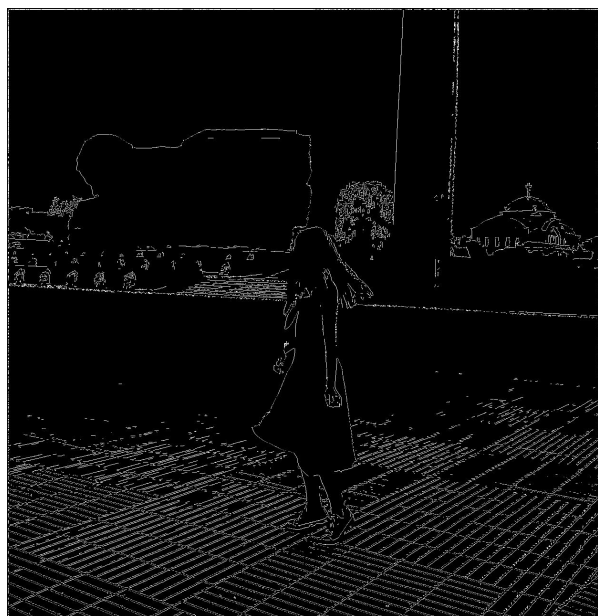
Изображение в полутонах



Выделение границ



Изображение в полутонах



Выделение границ



Изображение в полутонах



Выделение границ