

Санкт-Петербургский политехнический университет Петра Великого  
Институт машиностроения, материалов и транспорта  
Высшая школа автоматизации и робототехники

# Отчёт

по лабораторной работе №3

Дисциплина: Техническое зрение

Тема: Применение преобразования Фурье для фильтрации изображений

Студент гр. 3331506/70401

Архипов А.Е.

Преподаватель

Варлашин В.В.

«    » \_\_\_\_\_ 2020 г.

Санкт-Петербург

2020

## Задание

Рассмотреть применение преобразования Фурье для фильтрации изображений в частотной области.

## Задачи

- 1) Реализовать прямое дискретное преобразование Фурье (ДПФ), с возможностью вывода спектра, и обратное ДПФ.;
- 2) Реализовать фильтр низких и высоких частот в соответствии с вариантом задания;
- 3) Провести по отдельности свертку какого-либо изображения с ядром фильтров: Собеля, усредняющего, Лапласа;
- 4) Провести корреляцию изображений автомобильных номеров по очереди с 3-мя символами.

## Ход работы

### 1. Реализация дискретного преобразования Фурье

Дискретное преобразование Фурье реализовано методом *transformFourier*. Одним из параметров метода является флаг, с помощью которого можно выполнять прямое и обратное преобразование. Прямое преобразование Фурье выполняется при флаге равным 0, согласно формуле 1.

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}, \quad (1)$$

где  $f(x, y)$  – цифровое изображение размерами  $M \times N$ .

Помимо флага необходимо передать изображение в формате CV\_32FC1, а результат записывает в двухканальное изображение в формате CV\_32FC2. В нулевом канале находится вещественная часть результата, в первом – мнимая.

Обратное преобразование Фурье выполняется при флаге равным 1, согласно формуле 2.

$$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \quad (2)$$

Данный метод, помимо флага, принимает на вход двухканальное изображение в формате CV\_32FC2, а результат записывает в одноканальное изображение в формате CV\_32FC1.

Для оценки результата работы функции было проведено сравнение времени прямого и обратного преобразования Фурье собственной реализации с временем работы аналогичных функций из библиотеки OpenCV. Сравнение происходило в режиме Release.

Также было проведено сравнение результата с исходным полутоновым изображением и найдена среднеквадратичная погрешность. Для оценки использовалось изображение размером 100x100 пикселей. Результаты приведены в таблице 1.

Таблица 1 – Результаты сравнения

Время выполнения, мс		Среднеквадратичная погрешность	
Функции из библиотеки	Реализованные функции	Функции из библиотеки	Реализованные функции
77	6297	0	0

Получение спектра изображения было реализовано функцией *spectrum*. Данная функция работает следующим образом:

- 1) Разделение Фурье-образа на вещественную и мнимую части;
- 2) Вычисление амплитуды, согласно формуле 3;

$$|F(u, v)| = \sqrt{Re^2 + Im^2} \quad (3)$$

- 3) Перестановка квадрантов 1-3 и 2-4;
- 4) Переход к логарифмическому масштабу.

Пример работы показан на рисунке 1.

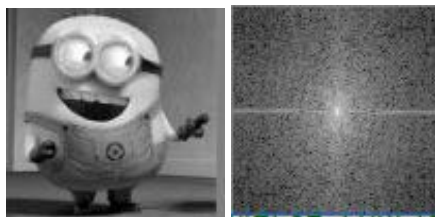


Рисунок 1 – Изображение и его спектр

## 2. Реализация фильтра низких и высоких частот

В соответствии с вариантом задания был реализован фильтр Баттерворта. Передаточные функции фильтров низких частот (ФНЧ) и фильтра высоких частот (ФВЧ) показаны на формулах 4 и 5 соответственно.

$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}} \quad (4)$$

$$H(u, v) = \frac{1}{1 + [D_0/D(u, v)]^{2n}} \quad (5)$$

где  $D_0$  — частота среза;  $D(u, v)$  — расстояние от центра частотного прямоугольника,  $n$  — порядок фильтра.

Фильтр Баттерворта реализован с помощью метода *ButterworthFilter(Mat& inputImage, Mat& outputImage, float D, int n, int flag)*. Данный метод работает следующим образом:

- 1) Заполнение фильтра;
- 2) Перестановка квадрантов изображения фильтра;
- 3) Вычисление Фурье-образа входного изображения;
- 4) Поэлементное перемножение изображения фильтра и Фурье-образа входного изображения;
- 5) Обратное ДПФ результата перемножения.

На рисунках 2 и 3 представлены: исходное изображение, отфильтрованный спектр изображения и результат фильтрации.

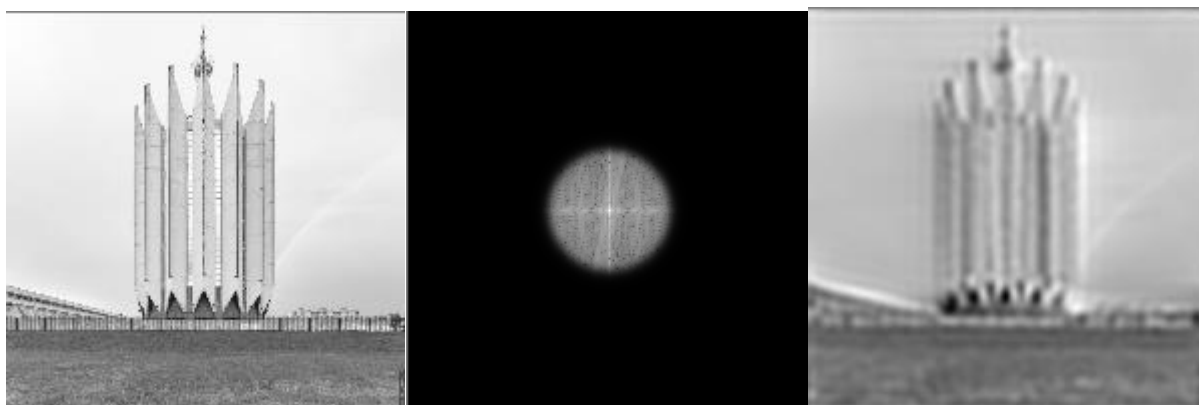


Рисунок 2 – Обработка ФНЧ

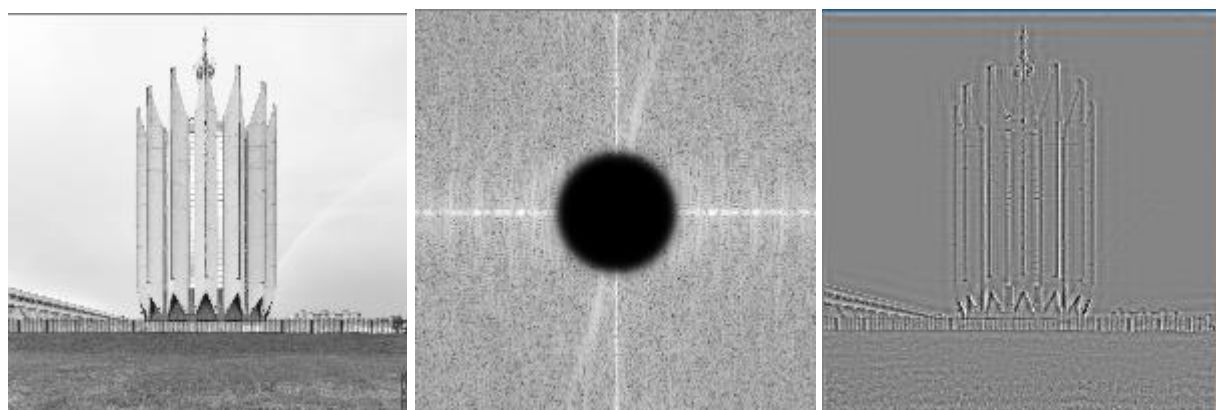


Рисунок 3 – Обработка ФВЧ

### 3. Свертка изображения с ядром различных фильтров

Для свертки изображения с различными ядрами фильтров был реализован метод *covolution(Mat& inputImage, Mat& outputImage, int flag)*. Если параметр *flag* равен 0, то производится свертка с ядром Лапласа.

На рисунке 4 представлены: спектр исходного изображения, спектр фильтра Лапласа, результат свертки.

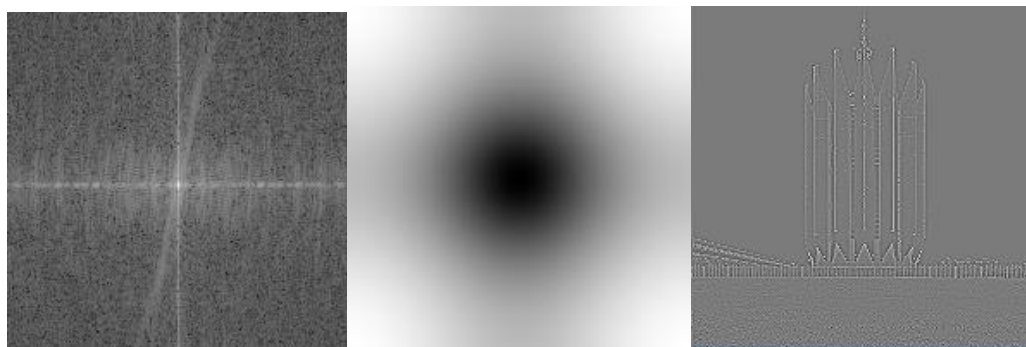


Рисунок 4 – Свертка с ядром фильтра Лапласа

Если параметр flag равен 1, то производится свертка с ядром Собеля по вертикали.

На рисунке 5 представлены: спектр исходного изображения, спектр фильтра Собеля по вертикали, результат свертки.

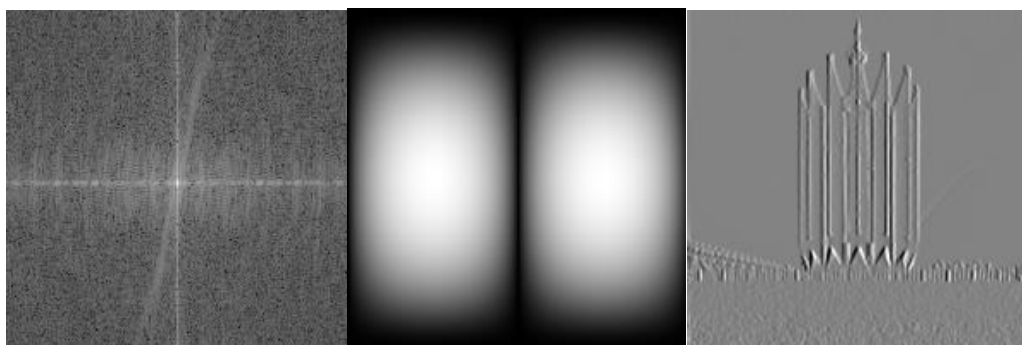


Рисунок 5 – Свертка с ядром фильтра Собеля по вертикали

Если параметр flag равен 2, то производится свертка с ядром Собеля по горизонтали.

На рисунке 6 представлены: спектр исходного изображения, спектр фильтра Собеля по горизонтали, результат свертки.

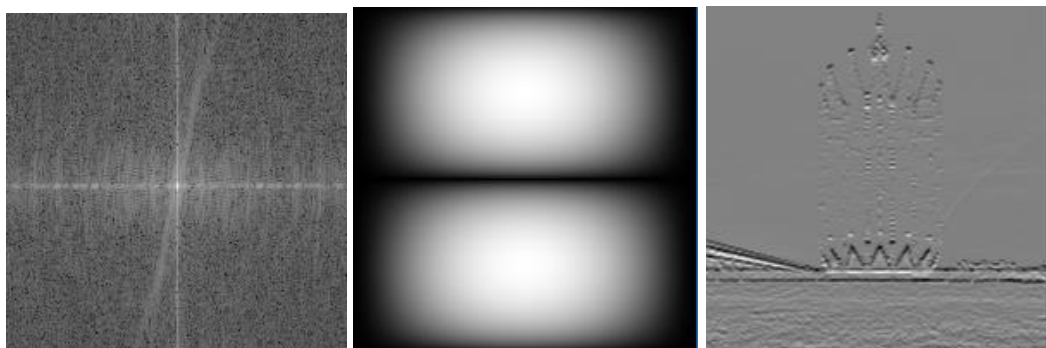


Рисунок 6 – Свертка с ядром фильтра Собеля по горизонтали

Если параметр `flag` равен 3, то производится свертка с ядром усредняющего фильтра.

На рисунке 7 представлены: спектр исходного изображения, спектр усредняющего фильтра, результат свертки.

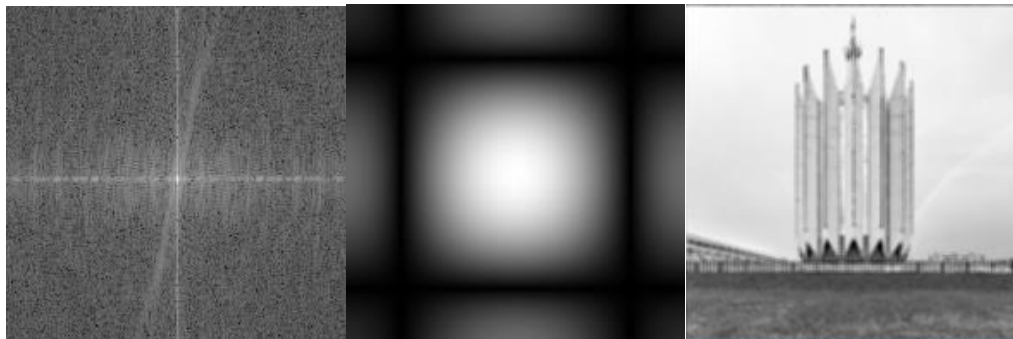


Рисунок 7 – Свертка с ядром усредняющего фильтра

#### 4. Поиск символов на автомобильном номере

Поиск символа на автомобильном номере реализован с помощью метода *findSymbol(Mat inputImage, Mat symbol)*. Алгоритм работы метода:

- 1) Вычитание среднего значения интенсивности из каждого значения интенсивности пикселей изображения и символа;
- 2) Изменение размеров изображения с номером до оптимальных;
- 3) Перенос изображения символа в левый верхний угол пустого изображения с оптимальными размерами;
- 4) Нахождение Фурье-образов изображения с номером и изображения с символом;
- 5) Корреляция данных Фурье-образов;
- 5) Обратное преобразование результата корреляции;
- 6) Обрезание результата обратного преобразования до оригинальных размеров;
- 7) Нормализация и нахождение порогового значения для пороговой фильтрации ( $maxVal - 0,03$ );
- 8) Пороговая фильтрация.

Результат работы метода представлен на рисунке 8.

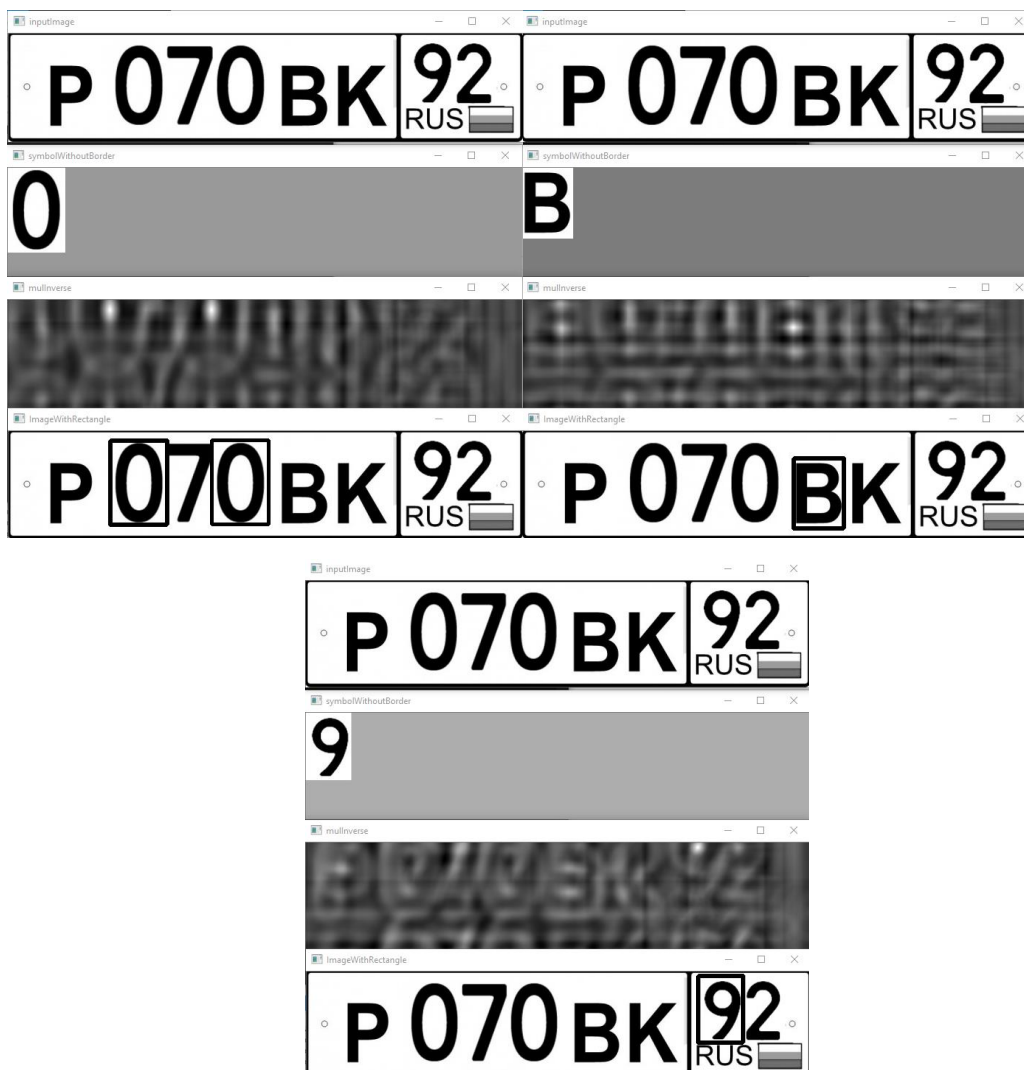


Рисунок 8 – Поиск символов на автомобильном номере

## Вывод

В ходе работы было изучены способы применения дискретного преобразования Фурье для фильтрации изображений в частотной области. Реализованы прямое и обратное преобразования Фурье, проведено сравнение с встроенной функцией. Различия в результатах обусловлены погрешностью вычислений. Также реализованы фильтры Гаусса низких и верхних частот. Проведена свертка изображения с ядром фильтров Собеля, Лапласа, усредняющего. Реализован метод поиска символа на изображении автомобильного номера.



## Защита

### 1. Сравнение фильтра Баттерворта и усредняющего фильтра

Усредняющий фильтр (box-filter) представляет собой низкочастотный фильтр. Идея применения фильтра достаточно ясна. Заменой исходных значений элементов изображения на средние значения по маске фильтра достигается уменьшение «резких» переходов уровней яркости.

Фильтр Баттерворта может быть как фильтром низких частот, так и фильтром высоких частот. Для сравнения с усредняющим фильтром будем использовать его как фильтр низки частот.

Для сравнения двух фильтров приведем их с общему виду, чтобы частота среза была одинаковой. Для этого можно воспользоваться функцией *getStructuringElement* из библиотеки OpenCV. Спектры фильтров, после применения которых имеют наименьшее среднеквадратичное отклонение, представлены на рисунке 9.

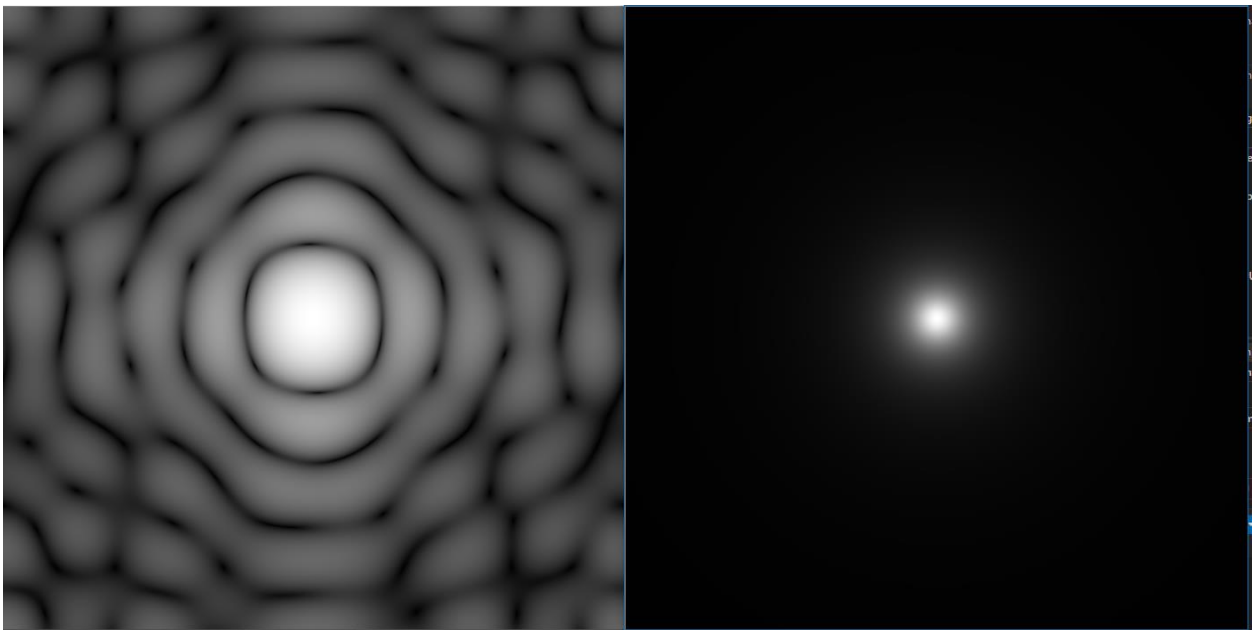


Рисунок 9 — Спектры усредняющего фильтра и фильтра Баттерворта  
Результат фильтрации представлен на рисунке 10.

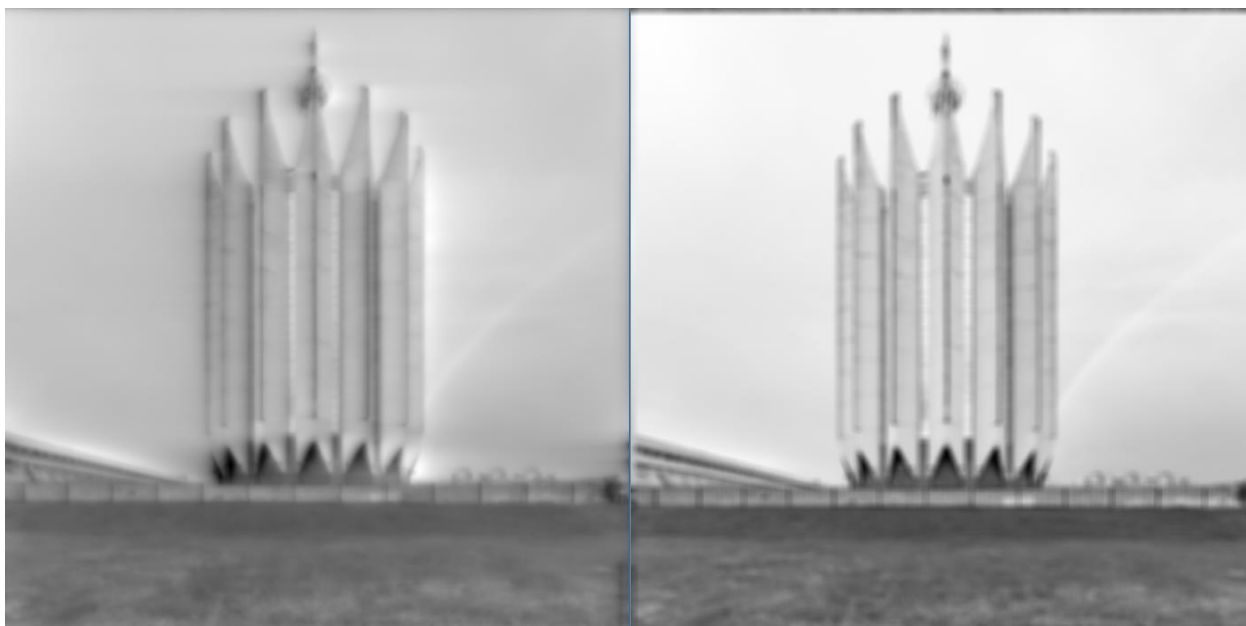


Рисунок 10 — Результат фильтрации фильтром Баттерворта и усредняющим фильтром

Среднеквадратичное отклонение составляет 33,4813

## 2. Обнаружение отверстий на номерном знаке

Для нахождения объекта на изображении используется операция корреляции. Для уменьшения влияния перепадов интенсивности пикселей выполняется вычитание среднего значения. Это значительно увеличивает качество обнаружения. Однако при попытке найти на номерном знаке автомобиля отверстие, оно обнаруживается совершенно не там, результат представлен на рисунке 11.



Рисунок 11 — Поиск отверстия только с вычитанием среднего

Для решения этой проблемы необходимо реализовать полноценную нормированную кросс-корреляцию, формула которой приведена на рисунке 12.

$$h[m,n] = \frac{\sum_{k,l} (g[k,l] - \bar{g})(f[m-k,n-l] - \bar{f}_{m,n})}{\left( \sum_{k,l} (g[k,l] - \bar{g})^2 \sum_{k,l} (f[m-k,n-l] - \bar{f}_{m,n})^2 \right)^{0.5}}$$

среднее изображение
средний фильтр

Рисунок 12 — Формула нормированной кросс-корреляции

После реализации нормированной кросс-корреляции, исходные изображения (символ и номерной знак) по сути превращаются в бинарные, представляющие из себя набор единиц с разными знаками. Пример такого изображения представлен на рисунке 13.

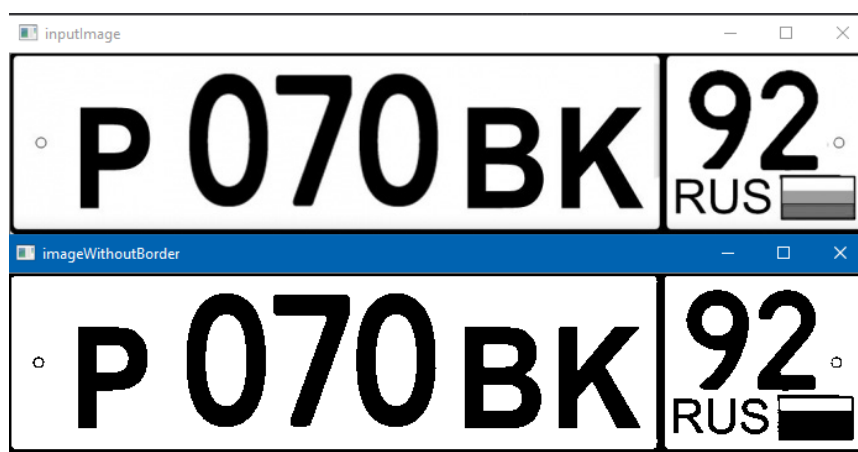


Рисунок 13 — Изображение после бинаризации

Результат работы нормированной кросс-корреляции приведен на рисунке 14.

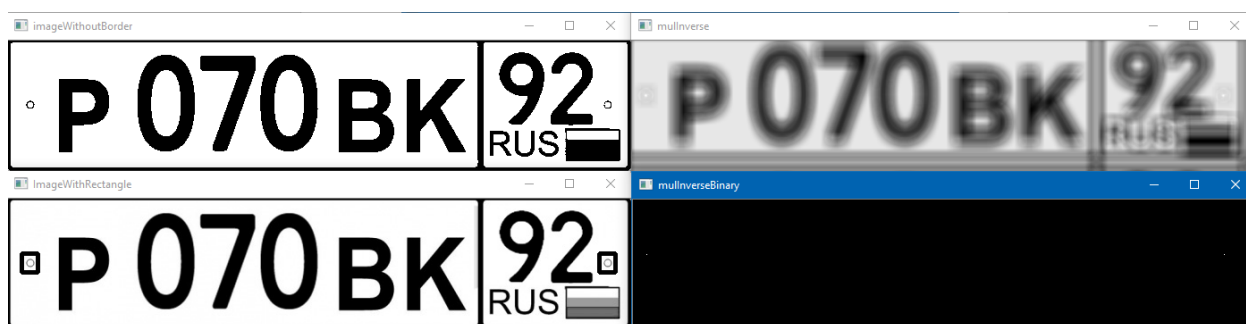


Рисунок 14 — Применение нормированной кросс-корреляции