

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

Преподаватель

Засецкий В.С.

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

Для реализации дискретного преобразования Фурье был написан класс *Fourier*.

Прямое дискретное преобразование Фурье реализовано методом *int forwardTransform()* согласно формуле

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

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

Данная функция использует одноканальное изображение в формате CV_32FC1, а результат записывает в атрибуты *m_imageRe* и *m_imageIm* — вещественная и мнимая часть результата соответственно.

Обратное дискретное преобразование Фурье реализовано методом *int inverseTransformFromImage()* согласно формуле

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

Данная функция принимает на вход вещественную и мнимую часть Фурье-образа, а результат записывает в одноканальное изображение в формате CV_32FC1.

Для оценки результата работы функции проведено сравнение времени последовательной работы обеих функций с временем работы аналогичных функций из библиотеки OpenCV, а также сравнение результата с исходным полутоновым изображением, и найдена среднеквадратичная погрешность. Результаты приведены в таблице 1 (конфигурация Release).

Таблица 1 — Результаты сравнения

| Время выполнения, с | | Среднеквадратичная погрешность | |
|----------------------|-----------------------|--------------------------------|-----------------------|
| Библиотечная функция | Реализованная функция | Библиотечная функция | Реализованная функция |
| 0,001 | 16,64 | $2,2 \cdot 10^{-7}$ | $5,5 \cdot 10^{-5}$ |

Получение спектра изображения было реализовано функцией *int showImageSpectre*. Данная функция берёт значение амплитуды, вычисленное в ходе прямого преобразования Фурье по формуле $|F(u, v)| = \sqrt{Re^2 + Im^2}$ и записанное в атрибут *m_spectre*, переводит к логарифмическому масштабу и меняет местами квадранты 1-3 и 2-4 с помощью функции *int swapDFT()*.

Пример выполнения дискретного преобразования Фурье и отображения спектра представлен на рисунке 1.



Рисунок 1 — Спектр и исходное изображение

В дальнейшем в работе используется библиотечная функция преобразования Фурье.

2. Реализация полосового частотного фильтра

В соответствии с вариантом задания был реализован режекторный фильтр Баттерворта, который задаётся следующим уравнением:

$$H(u, v) = \frac{1}{1 + \left(\frac{DW}{D^2 - D_0^2} \right)^{2n}},$$

где D — расстояние $D(u, v)$ от центра фильтра,

D_0 — частота среза,

W — ширина полосы,

n — порядок фильтра Баттерворта.

Режекторный фильтр Баттерворта реализован с помощью метода *int butterworthNotch(float D0, float W, float n)*. Аргументами являются параметры фильтра.

Алгоритм метода:

1. Получение спектра исходной картинки.
2. Создание экземпляра *Mat* того же размера, что и спектр, заполнение его коэффициентами фильтра согласно передаточной функции.
3. Обмен квадрантов фильтра.
4. Перемножение спектра исходной картинки и фильтра.
5. Обратное преобразование Фурье и вывод результата.

На рисунке 2 показаны исходное изображение, фильтр, выходное изображение, спектр до и после перемножения (слева направо сверху вниз).

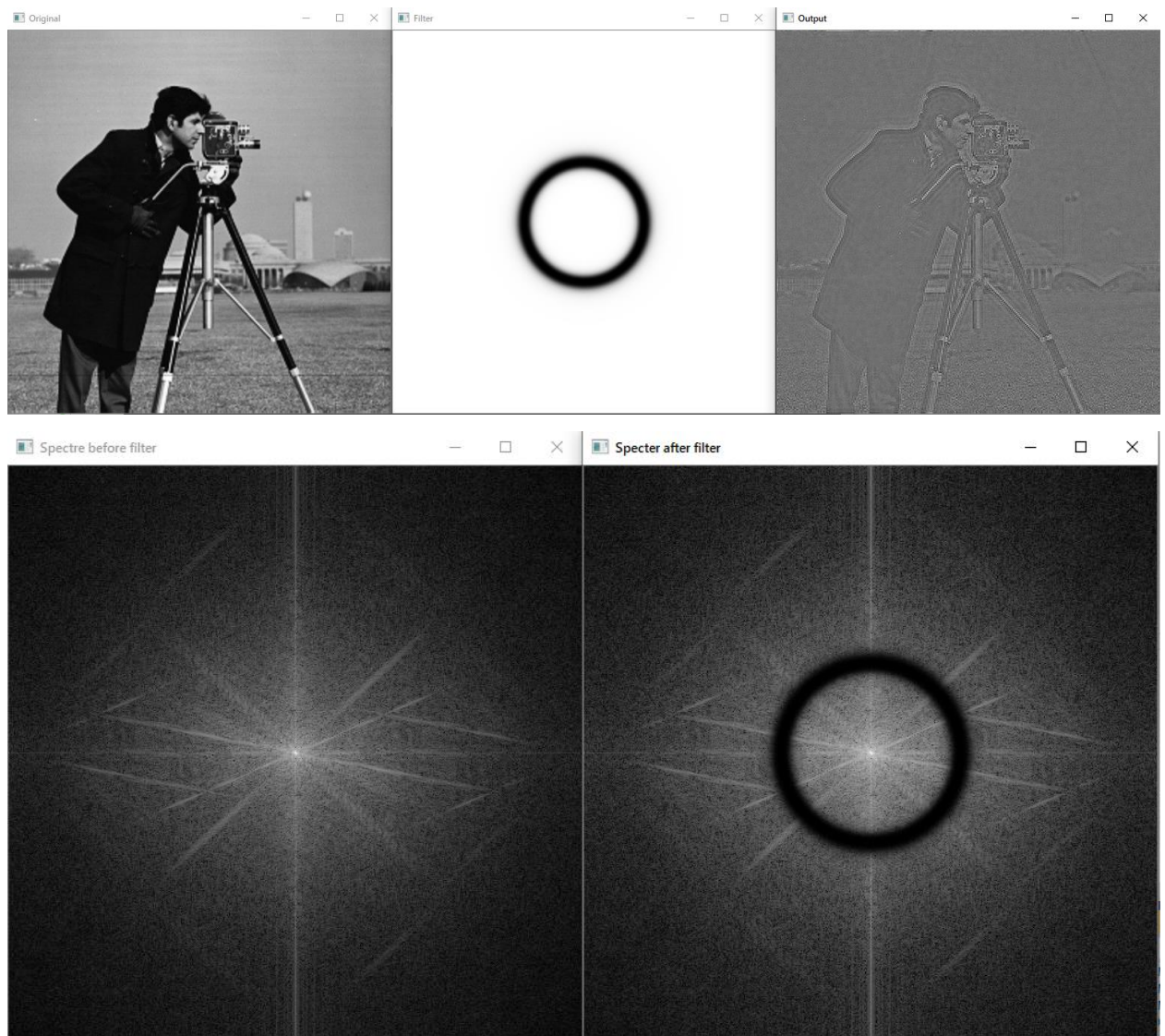


Рисунок 2 — Результат фильтрации ($D0 = 80$, $W = 20$, $n = 2$)

На рисунках ниже изображены результаты фильтрации с различными значениями частоты среза и ширины полосы.



Рисунок 3 — Результат фильтрации ($D0 = 40$, $W = 20$, $n = 2$)

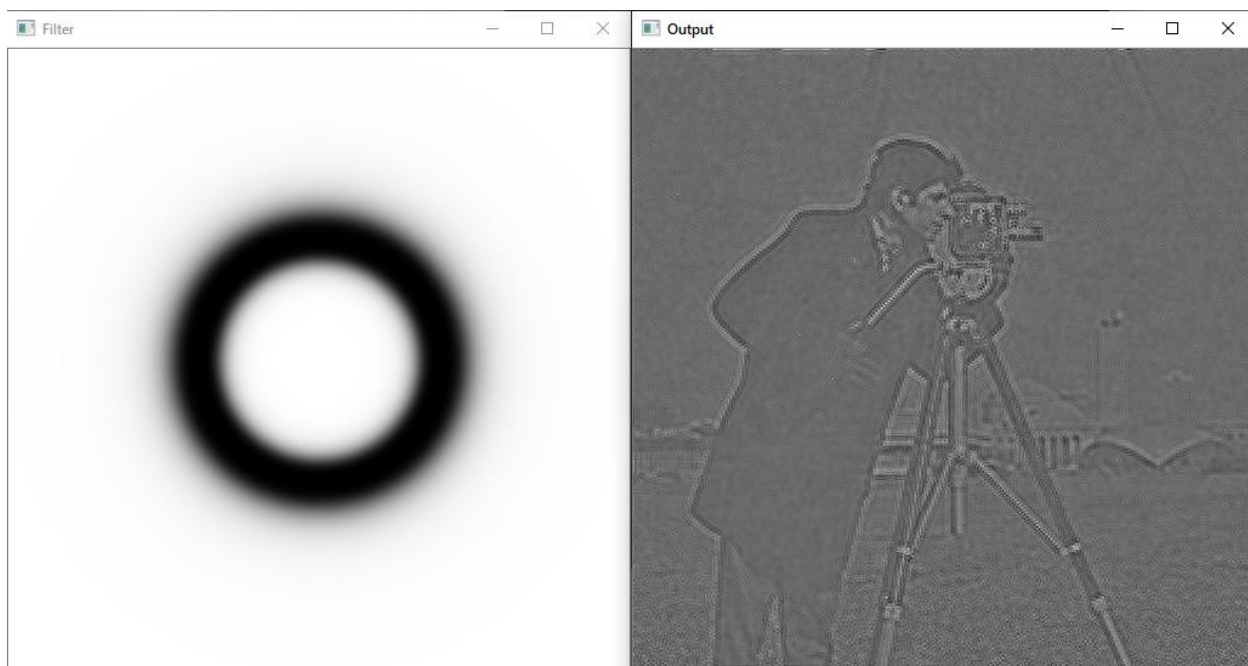


Рисунок 4 — Результат фильтрации ($D0 = 100$, $W = 50$, $n = 2$)

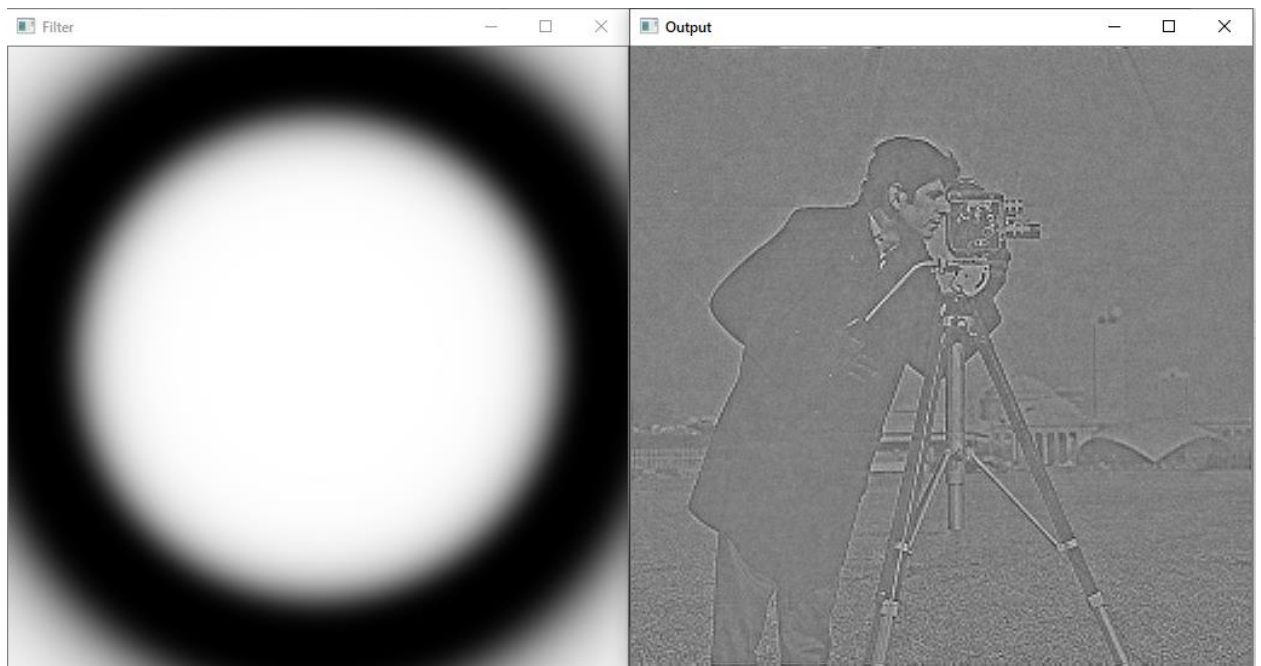


Рисунок 5 — Результат фильтрации ($D0 = 240$, $W = 110$, $n = 2$)

3. *Свёртка изображения с различными фильтрами*

Согласно теореме о свёртке, свёртка изображений эквивалентна перемножению их Фурье-образов.

В работе свёртка изображения с фильтром Собеля (вертикальный и горизонтальный), с усредняющим фильтром и с фильтром Лапласа реализована методами *void sobel (bool isHoriz)*, *void laplace()*, *void box()*.

Алгоритм методов следующий:

1. Создание экземпляра *Mat* 3×3 и заполнение коэффициентами фильтра.
2. Вызов вспомогательного метода *void applyFilter(Mat& filter)*, в котором выполняются дальнейшие шаги.
3. Нахождение оптимального размера холста для преобразования Фурье и расширение исходного изображения и фильтра до полученного размера.
4. Выполнение прямого преобразования Фурье для изображения и фильтра.
5. Комплексное перемножение Фурье-образов изображения и фильтра.

6. Выполнение обратного преобразования Фурье для результата перемножения.
7. Восстановление исходных размеров.

На рисунке 6 представлены: спектр фильтра Собеля по горизонтали, спектр после перемножения, результат свёртки.

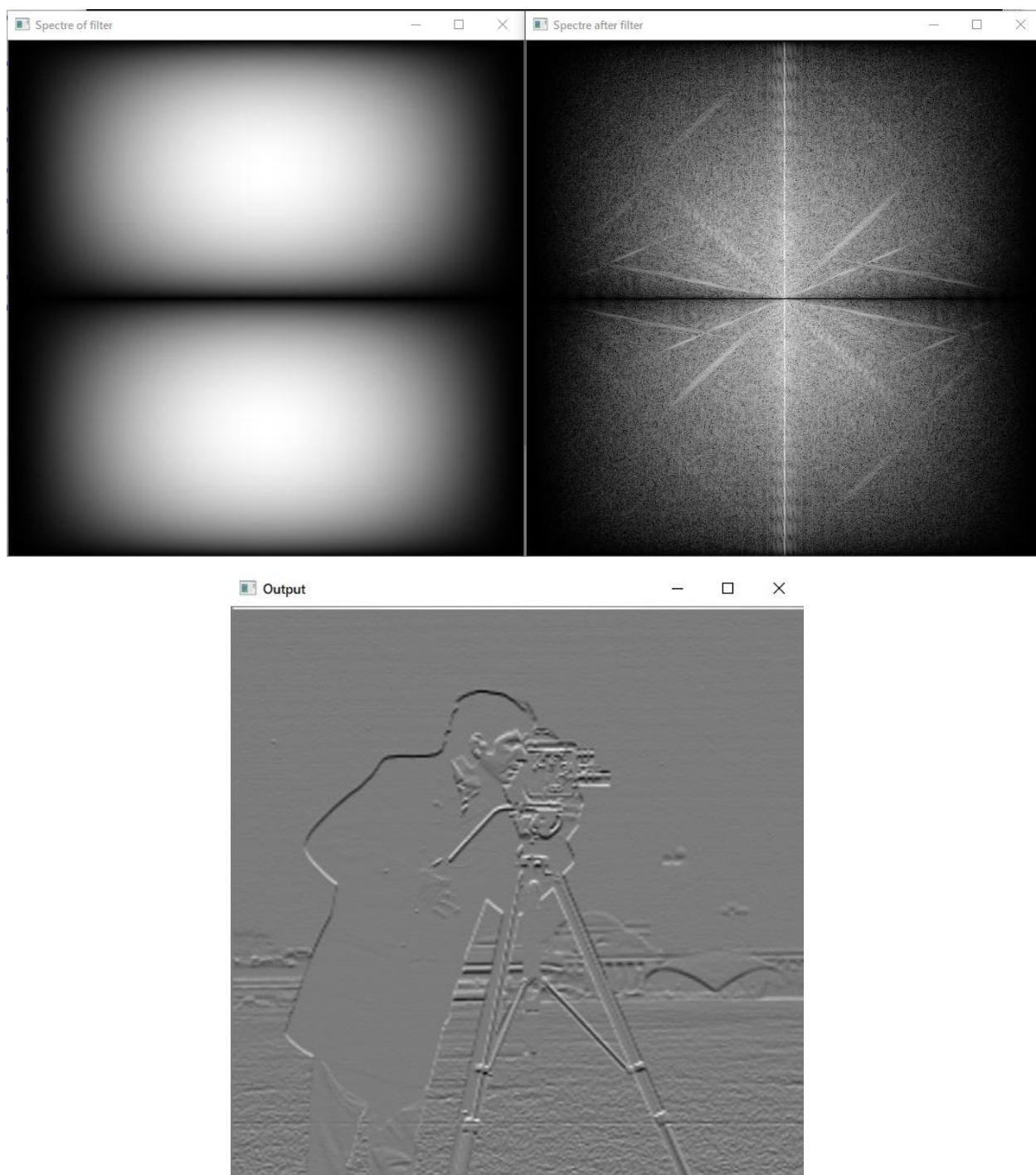


Рисунок 6 — Свёртка с фильтром Собеля по горизонтали

На рисунке 7 представлены: спектр фильтра Собеля по вертикали, спектр после перемножения, результат свёртки.

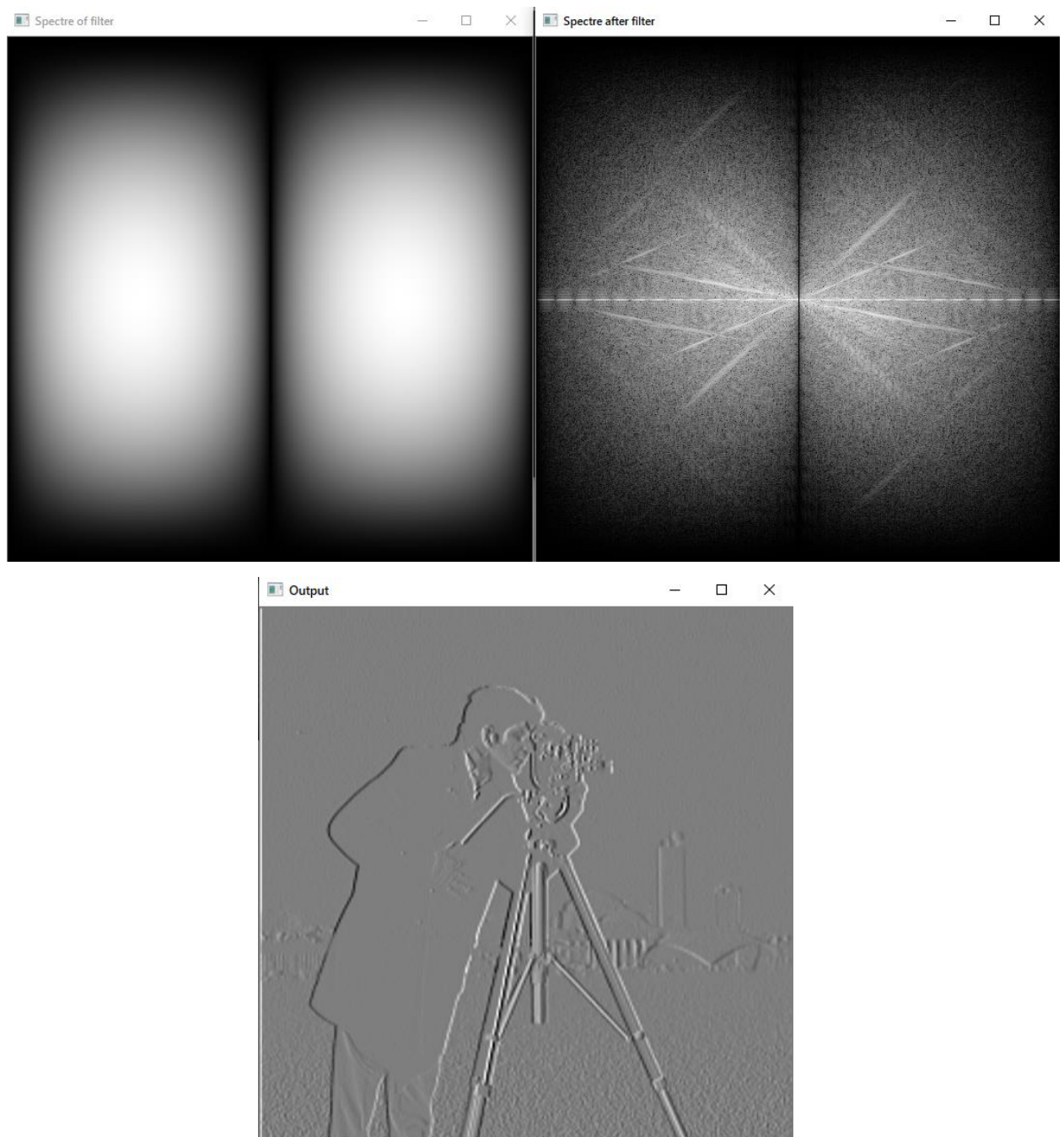


Рисунок 7 — Свёртка с фильтром Собеля по вертикали

На рисунке 8 представлены: спектр усредняющего фильтра, спектр после перемножения, результат свёртки.



Рисунок 8 — Свёртка с усредняющим фильтром

На рисунке 9 представлены: спектр фильтра Лапласа, спектр после перемножения, результат свёртки.

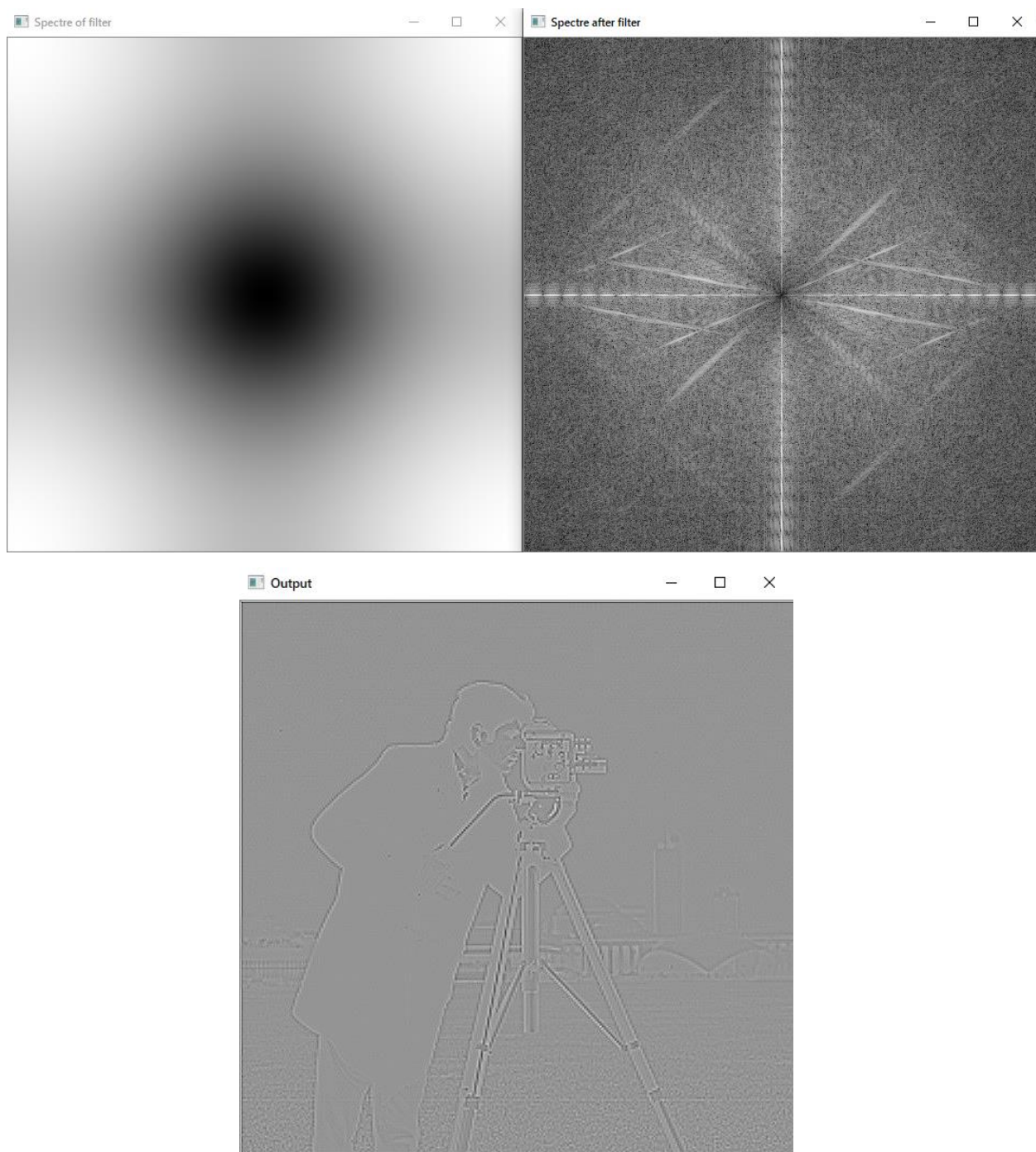


Рисунок 9 — Свёртка с фильтром Лапласа

4. *Поиск символов на автомобильном номере*

Поиск символа осуществляется при помощи корреляции образца (символа) с исходным изображением. Согласно теореме о корреляции, корреляция двух изображений эквивалентна произведению Фурье-образа одного из изображений на комплексно-сопряжённый Фурье-образ второго изображения.

В работе корреляция исходного изображения с символом реализована методом *int correlateWith(Mat& sample)*.

Алгоритм метода:

1. Вычисление оптимального размера для преобразования Фурье
2. Создание двух экземпляров Mat полученного размера, помещение номеров и символа в них.
3. Выполнение прямого преобразования Фурье для номеров и символа.
4. Создание экземпляра Mat для результатов корреляции, вычисление корреляции между номерами и символом.
5. Выполнение обратного преобразования Фурье для результата корреляции.
6. Восстановление исходного размера для корреляции.
7. Вычисление максимального значения интенсивности.
8. Выполнение пороговой фильтрации (порог — 0,99 от максимального значения).

На рисунках 10 — 12 представлены результаты поиска символов на автомобильных номерах. Сверху помещены сами номера, по середине — результат корреляции (после обратного преобразования Фурье), внизу — изображение после пороговой фильтрации и искомый символ.



Рисунок 10 — Поиск символа «С»



Рисунок 11 — Поиск символа «5»



Рисунок 12 — Поиск символа «7»

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 различием в алгоритмах. Встроенный использует быстрое преобразование Фурье. Также реализован режекторный фильтр Баттерворта. Проведена свертка изображения с ядром фильтров Собеля, Лапласа и усредняющего фильтра. Реализован метод поиска символа на изображении автомобильного номера.