

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

Соколов Д.А.

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

Для реализации дискретного преобразования Фурье был написан класс `Fourier2D`. Его UML диаграмма представлена на рисунке 1.

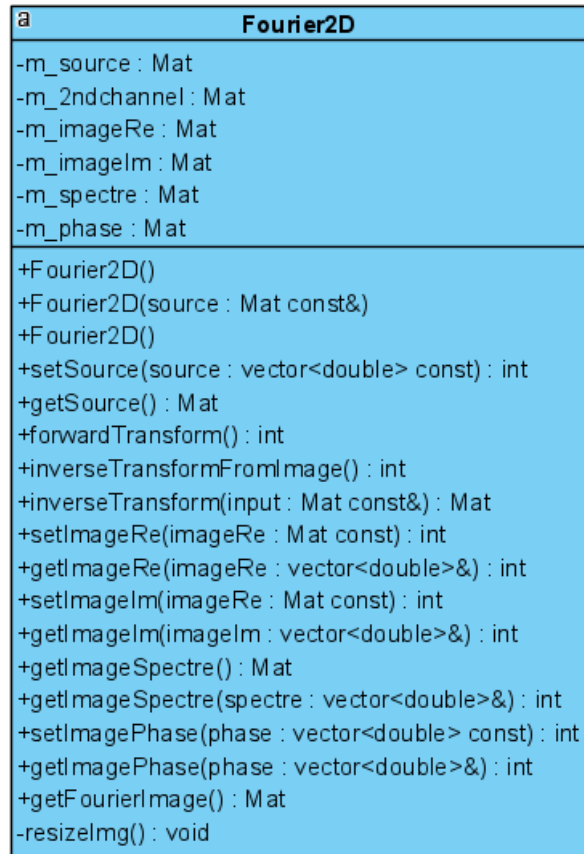


Рисунок 1 – UML диаграмма класса `Fourier2D`

Прямое дискретное преобразование Фурье реализовано методом `int forwardTransform()`, 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  $M \times N$ .

Данная функция использует одноканальное 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 дискретное преобразование Фурье реализовано методом *inverseTransform*, согласно формуле 2.

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

Данная функция 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 последовательной работы обеих функций с временем работы аналогичных функций из библиотеки OpenCV.

Также было проведено сравнение результата с исходным полутоновым изображением и найдена среднеквадратичная погрешность. Для оценки использовалось изображение размером 256x256 пикселей. Результаты приведены в таблице 1.

Таблица 1 – Результаты сравнения

Время выполнения, с		Среднеквадратичная погрешность	
Функции из библиотеки	Реализованные функции	Функции из библиотеки	Реализованные функции
0.00165778	160.223	0	0,005

Получение спектра изображения было реализовано функцией `cv::Mat createSpectrumHumanRdbl(const cv::Mat& src)`. Данная функция работает следующим образом:

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

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

Данная задача выполняется встроенной функцией `cv::magnitude()`

- 3) Перестановка квадрантов 1-3 и 2-4;

Выполняется функцией `void swapSpektr(cv::Mat &magI)`

4) Переход к логарифмическому масштабу.

Пример работы показан на рисунке 2.



Рисунок 2 – Исходное 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.

Фильтр Баттерворта реализован функцией `cv::Mat butterworth(size_t p, size_t q, size_t d_0, size_t n = 1, ButterworthFilter flag = HIGHPASS)`. Данная функция выполняет построение изображения фильтра в соответствии с передаточной функцией фильтра;

На рисунках 3 и 4 представлены: исходное изображение, спектр соответствующего фильтра до перестановки квадрантов и результат фильтрации.

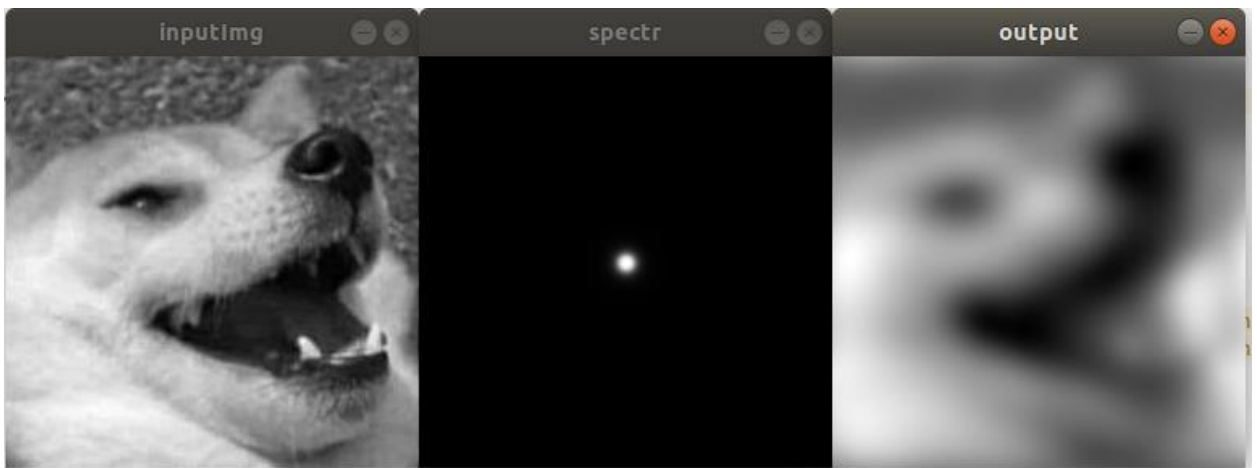


Рисунок 3 – Обработка ФНЧ

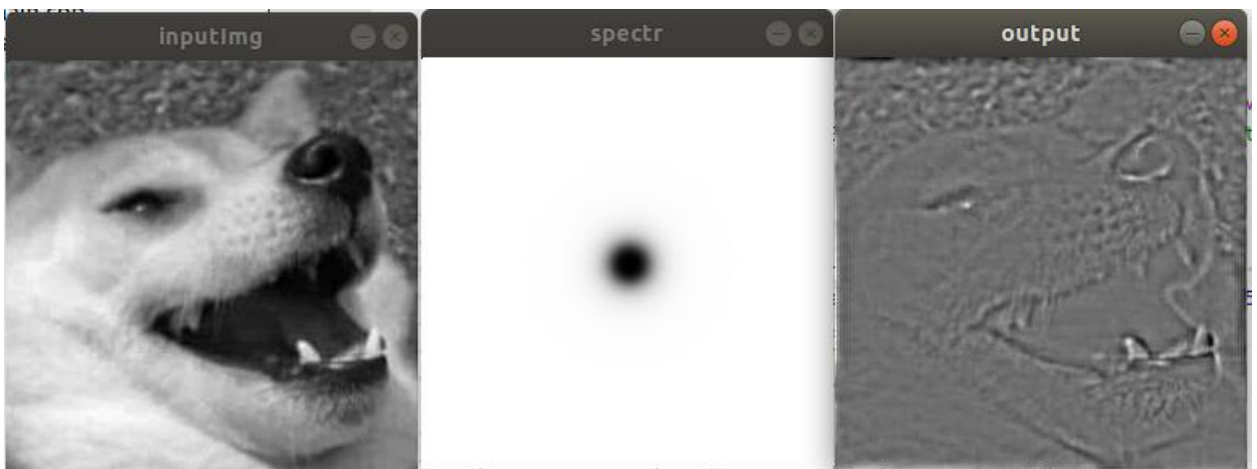


Рисунок 4 – Обработка ФВЧ

### 3. Свертка изображения с ядром различных фильтров

Для свертки изображения с ядрами различных фильтров, была написана функция `void fourierProcessing (cv::Mat filter, cv::Mat& src, cv::Mat& out, cv::Mat* spectr = nullptr)`. Она принимает ядро фильтра, входное и выходное изображения, и дополнительно контейнер для спектра фильтра.

На рисунке 5 представлены: исходное изображение, спектр исходного изображения, спектр фильтра Собеля по горизонтали, результат свертки.

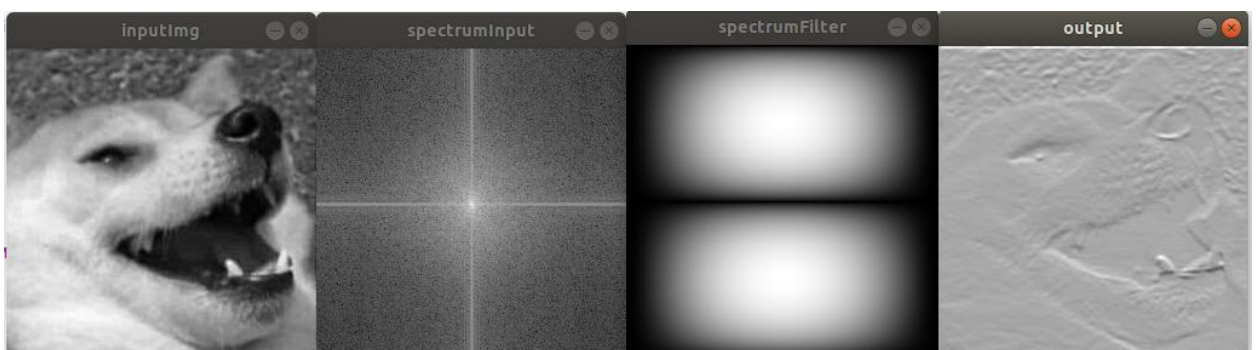


Рисунок 5 – Свертка с ядром фильтра Собеля по горизонтали

На рисунке 6 представлены: исходное изображение, спектр исходного изображения, спектр фильтра Собеля по вертикали, результат свертки.

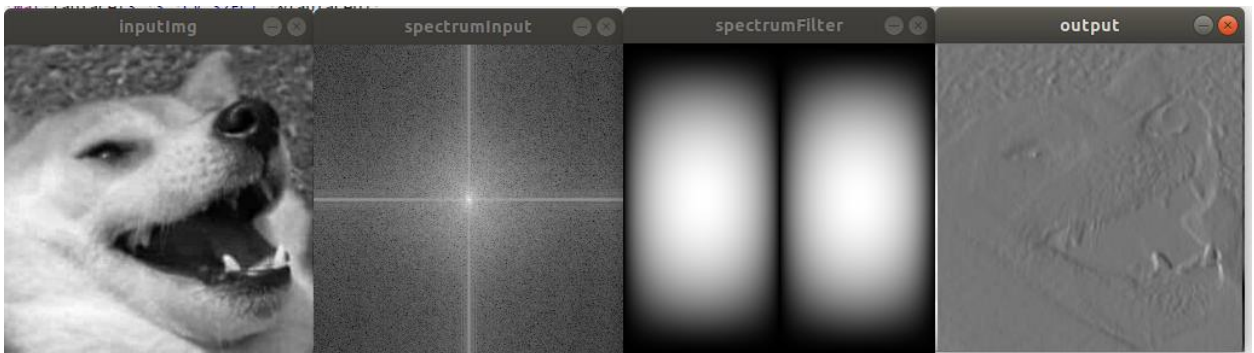


Рисунок 6 – Свертка с ядром фильтра Собеля по вертикали

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

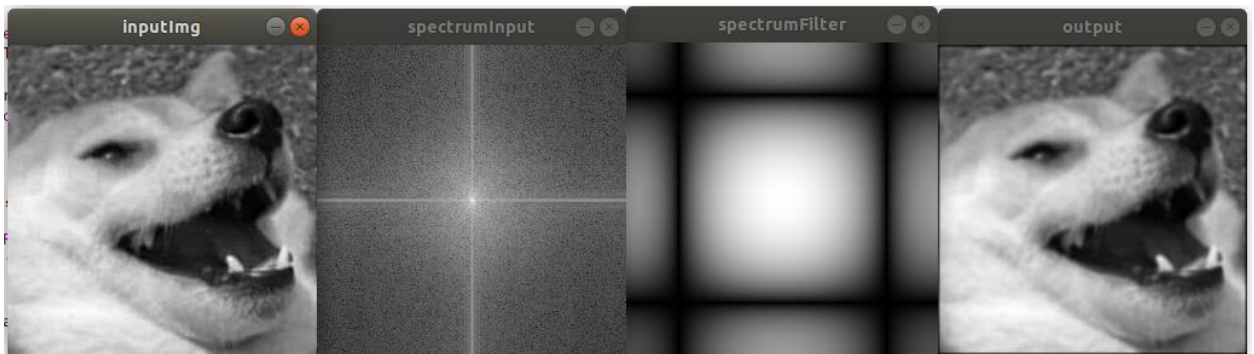


Рисунок 7 – Свертка с ядром усредняющего фильтра

На рисунке 8 представлены: исходное изображение, спектр фильтра Лапласа, результат свертки. Другое изображение выбрано для наглядности результатов.

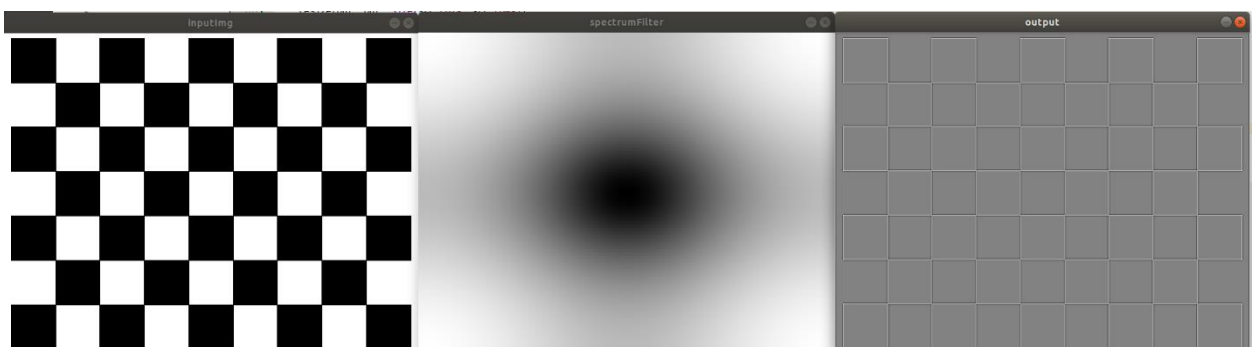


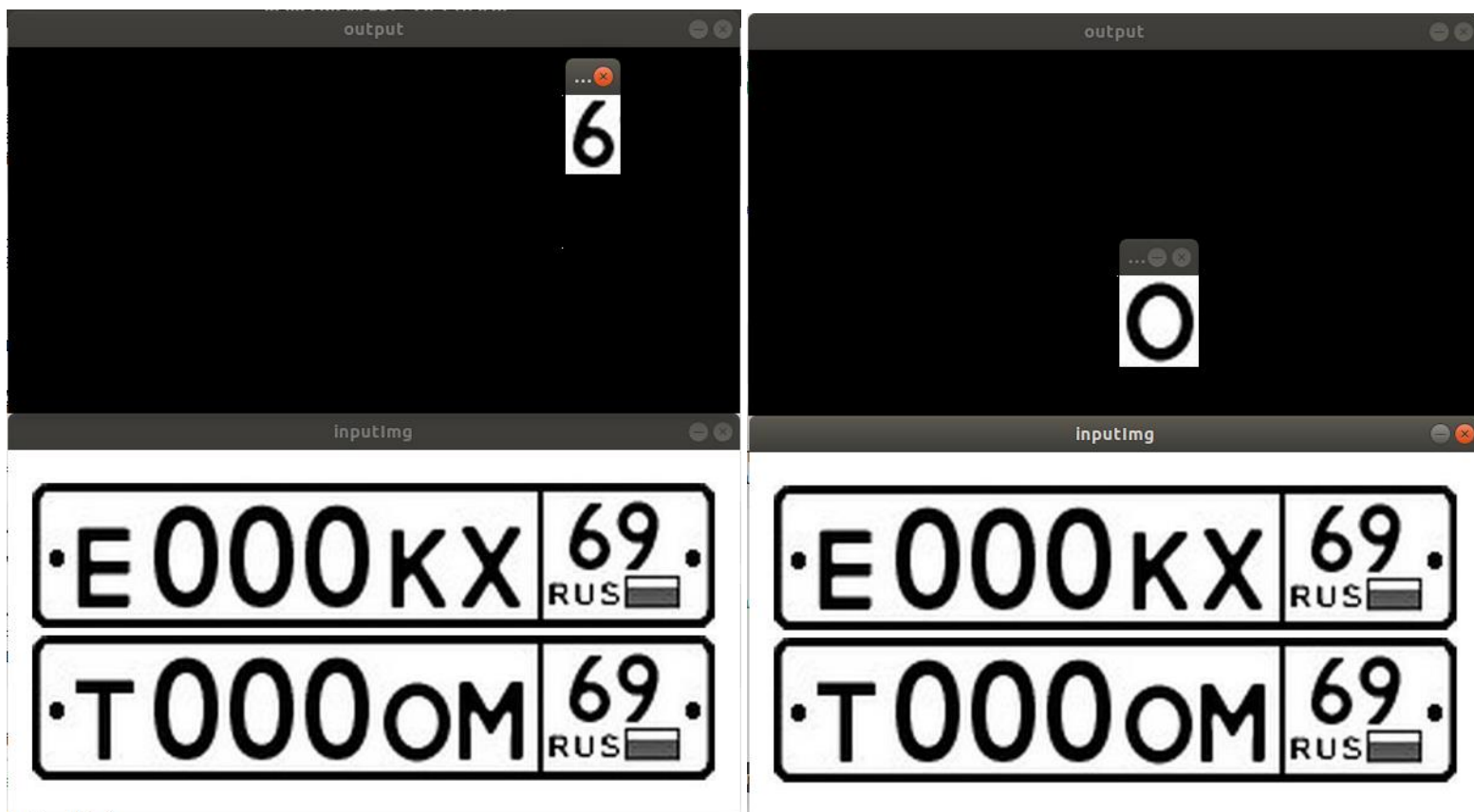
Рисунок 8 – Свертка с ядром фильтра Лапласа

#### 4. Поиск символов на автомобильном номере

Поиск символа на автомобильном номере реализован функцией `void fourierCorrelation (cv::Mat image, cv::Mat object, cv::Mat& out)`. Данная функция работает следующим образом:

- 1) Изменение размеров изображения с номером до оптимальных;
- 2) Перенос изображения символа в левый верхний угол пустого изображения с оптимальными размерами;
- 3) Нахождение Фурье-образов изображения с номером и изображения с символом;
- 4) Корреляция данных Фурье-образов с помощью встроенной функции `mulSpectrums`;
- 5) Обратное преобразование результата корреляции;
- 6) Обрезание результата обратного преобразования до оригинальных размеров;
- 7) Нормализация и нахождение порогового значения для пороговой фильтрации ( $\maxVal - 0.005$ );
- 8) Пороговая фильтрация.

Результат работы функции представлен на рисунке 9.





### **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 реализованы фильтры Баттерворта низких и верхних частот. Проведена свертка изображения с ядром фильтров Собеля, Лапласа, усредняющего. Реализован метод поиска символа на изображении автомобильного номера.