

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

Якименко Г.К.

Варлашин В.В.

« » _____ 2020 г.

Санкт-Петербург

2020

Задание

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

1. Прямое и обратное ДПФ

Прямое дискретное преобразование Фурье реализовано методом `void doDFT()`, 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 принимает одноканальное изображение в формате CV_32FC1, а результат записывает в двухканальное изображение в формате CV_32FC2, где нулевой канал служит для вещественной части образа, а первый для мнимой.

Обратное дискретное преобразование Фурье реализовано методом `void doIFT()`, согласно формуле:

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

Данный метод принимает на вход двухканальное изображение в формате CV_32FC2, а результат записывает в одноканальное изображение в формате CV_32FC1.

Для оценки результата работы функций, было решено сравнить их время работы с временем работы аналогичных функций библиотеки OpenCV, на примере одинакового изображения (200x131 пикселей). Результат сравнения приведен в таблице 1.

Таблица 1

Функция	dft (прям.)	dft (обр)	doDFT()	doIFT()
t, с	0.003	0.007	19.2	20.1

Получение спектра изображения было реализовано методом `void makeSpec(Mat src, Mat& dst)`, где `src` – двухканальное изображение, `dst` – одноканальное. Данный метод работает следующим образом:

- 1) Вычисление амплитуды по формуле:

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

2) Перестановка квадрантов dst (1-3, 2-4), с помощью функции void swapSpec(Mat& src);

3) Переход к логарифмическому масштабу;

4) Нормализация изображения от 0 до 1;

Исходное изображение, а также результат работы функций doDFT, makeSpec и doIFT, представлены на рисунке 1.



Рисунок 1 – исходное изображение, его спектр и изображение, полученное обратным преобразованием Фурье соответственно

2. Фильтр Баттерворта

Передаточная функция низких частот:

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

Передаточная функция высоких частот:

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

где D_0 – частота среза, $D(u, v)$ – расстояние от центра частотного прямоугольника.

Фильтр Баттерворта был реализован методом void bFilter(int n, FilterType filter_type), где n соответствует n передаточной функции, filter_type определяет передаточную функцию (LOW_PASS || HIGH_PASS).

Результат работы ФНЧ при $n = 1$ и $n = 5$ представлен на рисунках 2 и 3 соответственно.

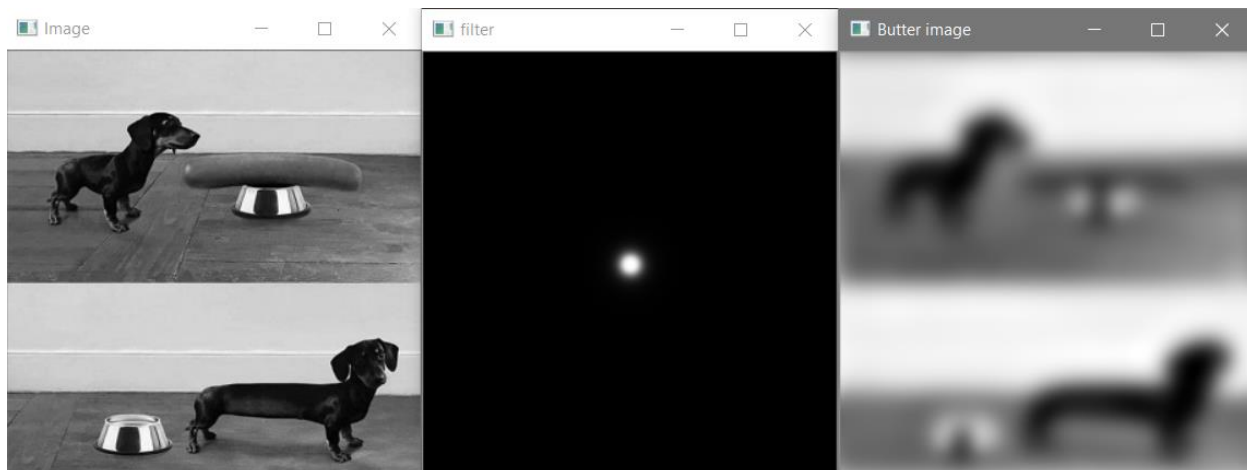


Рисунок 2 – результат работы ФНЧ, при $n = 1$

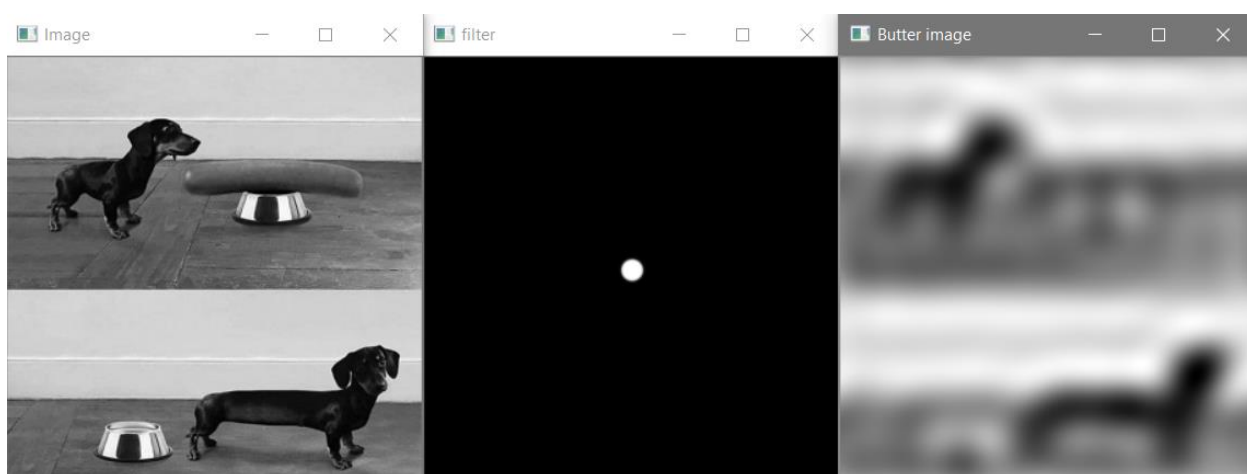


Рисунок 3 – результат работы ФНЧ, при $n = 5$

Результат работы ФВЧ, при $n = 1$ и $n = 5$, представлен на рисунках 3 и 4 соответственно.

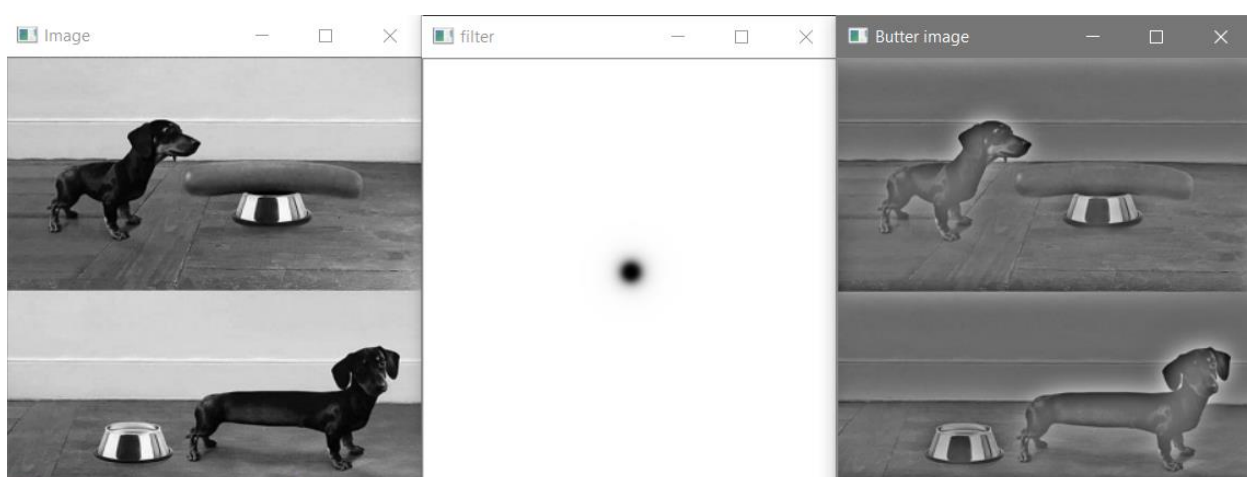


Рисунок 4 – результат работы ФВЧ, при $n = 1$

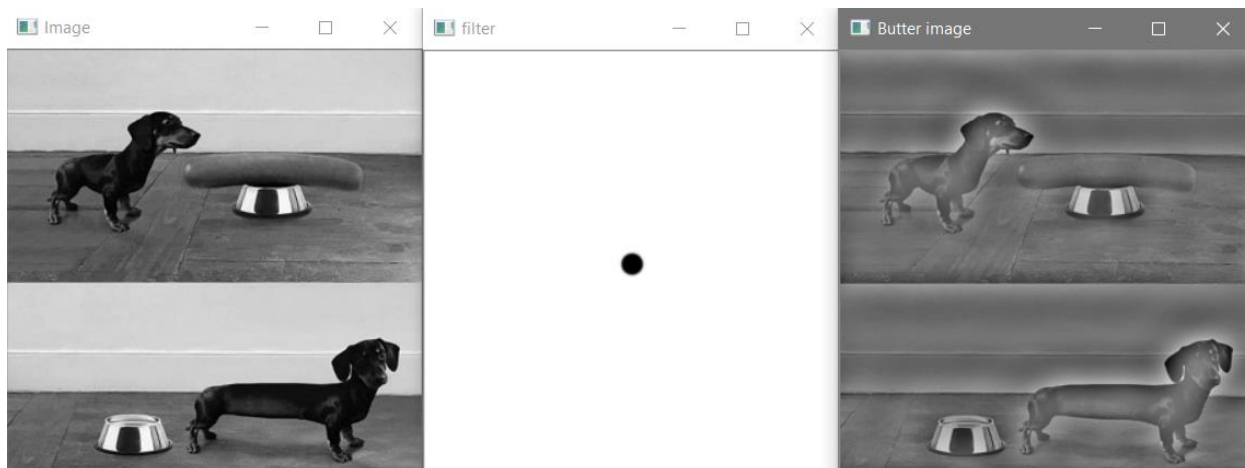


Рисунок 5 – результат работы ФВЧ, при $n = 5$

3. Свертка изображения с ядром различных фильтров

Свертка изображений реализована с помощью метода `void conv(ConvType conv_type)`, где `conv_type` – соответствует желаемому фильтру: `BOX`, `LAPLAS`, `SOBEL_H`, `SOBEL_L`.

Результат свертки с ядром усредняющего фильтра, представлен на рисунке 6 (оригинал, спектр свертки, результирующее изображение).

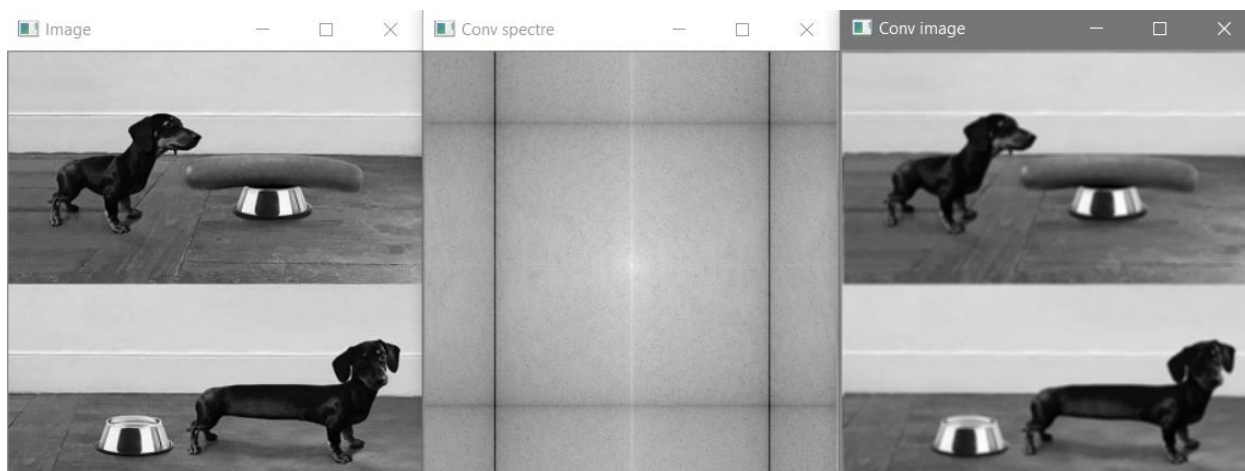


Рисунок 6 – результат свертки с усредняющим фильтром

Результат свертки с ядром фильтра Лапласа, представлен на рисунке 7 (оригинал, спектр свертки, результирующее изображение).

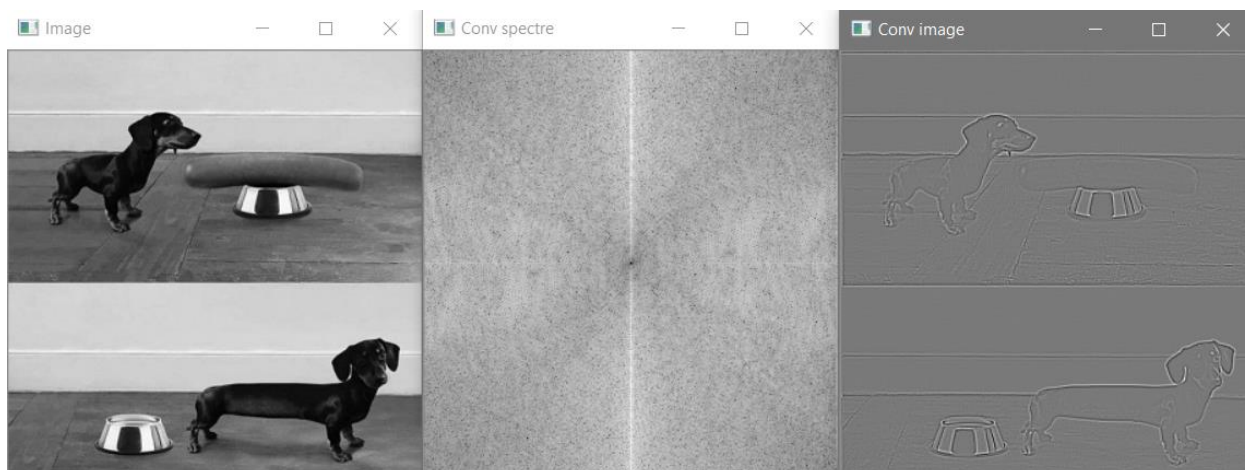


Рисунок 7 – результат свертки с фильтром Лапласа

Результат свертки с ядрами фильтров Собеля, представлены на рисунках 8 и 9 (оригинал, спектр свертки, результирующее изображение).

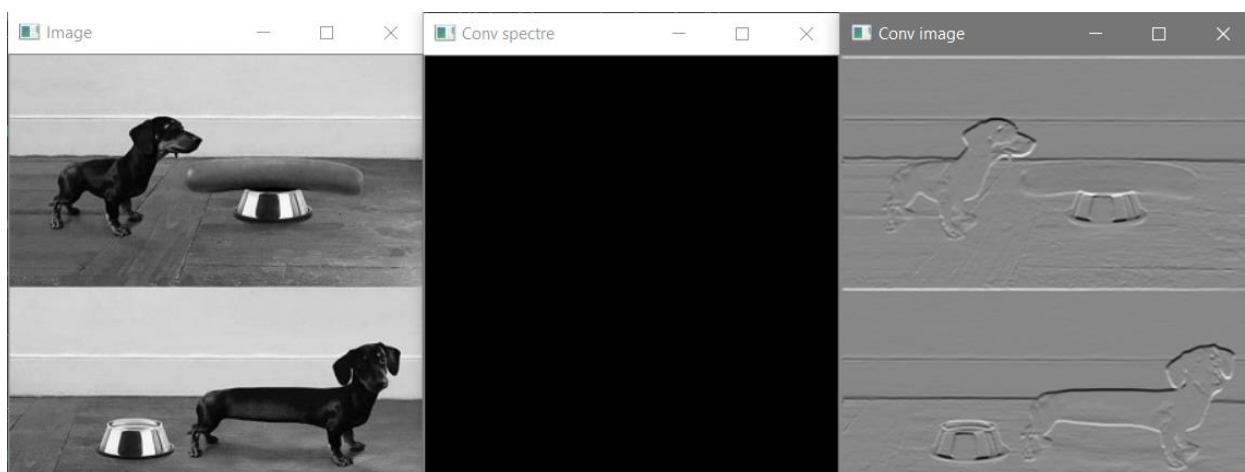


Рисунок 8 результат свертки с горизонтальным фильтром Собеля

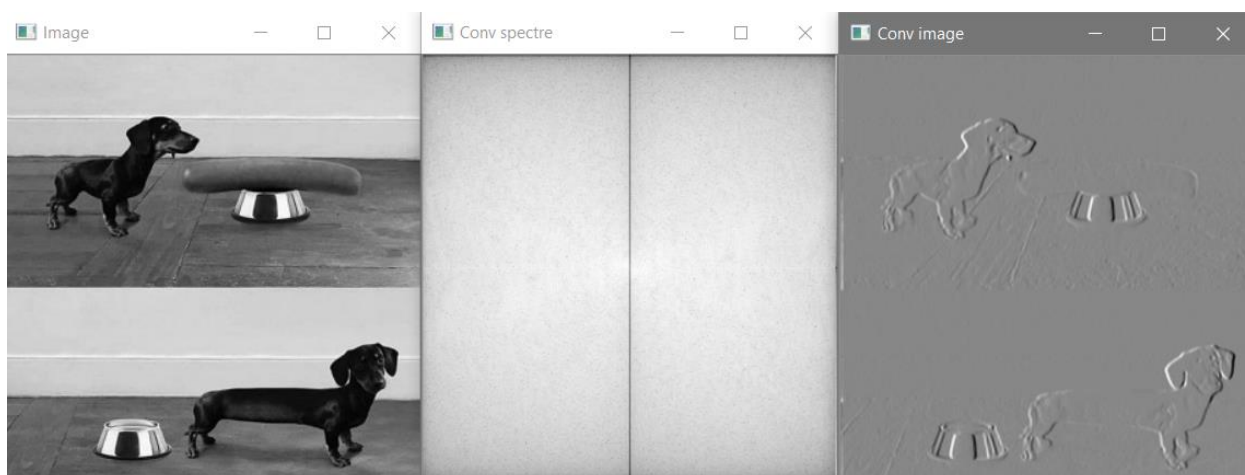


Рисунок 9 – результат свертки с вертикальным фильтром Собеля

4. Поиск паттернов на изображении

Поиск паттернов был реализован в методе `patternSearch(Mat image, Mat pattern)`, где `image` – изображение, `pattern` – искомый фрагмент. Метод работает следующим образом:

- 1) Производит расширение «холста» изображений до оптимальных (т.е. информативная часть изображений не увеличивается);
- 2) Выполняет ДПФ обоих изображений;
- 3) Производит корреляцию Фурье-образов изображений, с помощью встроенной функции `mulSpectrum`s;
- 4) Производит обратное ДПФ;
- 5) Обрезает неинформативную часть изображений.

Результат работы функции поиска паттернов: «А», «7», «0», на изображении представлен на рисунке 10.

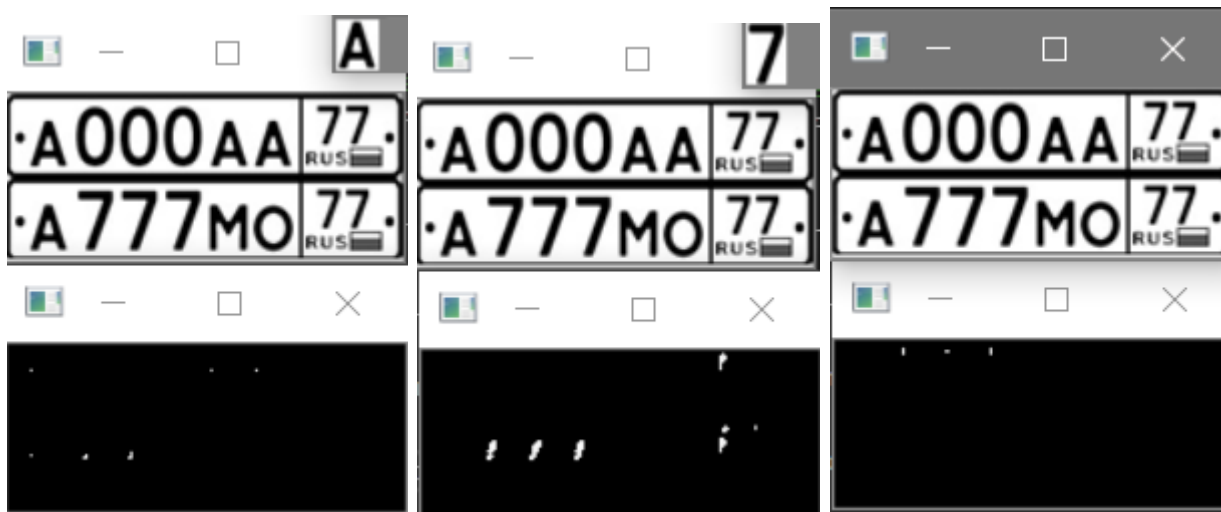


Рисунок 10 – поиск паттернов на изображении

Вывод

В ходе работы были изучены способы применения ДПФ для фильтрации изображений в частотной области. Реализованы прямое и обратное ДПФ, проведено сравнение со встроенной функцией. Различия в результатах обусловлены различием в алгоритмах: встроенная функция использует быстрое преобразование Фурье. Также реализованы фильтры Баттерворта верхних и нижних частот. Проведена свертка изображения с ядром усредняющего фильтра, горизонтального и вертикального фильтра Собеля, фильтра Лапласа. Реализован примитивный метод поиска паттернов на изображении.