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 обработки изображений

Студент гр. 3331506/70401 Шкабара Я. А.

Преподаватель Варлашин В. В.

« »_____2020 г.

Санкт-Петербург

2020

Оглавление

Задачи

1. Релизовать прямое и обратное преобразования Фурье;
2. Реализовать фильтр Гаусса для высоких и низких частот;
3. Произвести свёртку изображения с ядром фильтров: Собеля (по горизонтали и вертикали), усредняющего, Лапласа
4. Провести корреляцию (сравнение) изображений автомобильных номеров по очереди с 3-мя символами.

1. Преобразование Фурье

Прямое дискретное преобразование Фурье имеет вид:

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

где $f(x,y)-$ цифровое изображение размерами $M\times N$; $u = 0, 1, 2 \dots, M-1$; $v = 0, 1, 2 \dots, N-1$.

Прямое преобразование реализовано функцией *dDFT*, принимающей входное изображение в формате CV_32FC1. После преобразования результат записывается в изображение типа CV_32FC2.

Обратное дискретное преобразование Фурье имеет вид:

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

где $x = 0, 1, 2 \dots, M-1;\ y = 0, 1, 2 \dots, N-1$.

Обратное преобразование реализовано функцией *iDFT*, принимающей входное изображение в формате CV_32FC2. После преобразования результат записывается в изображение типа CV_32FC1.

На рисунке 1 представлен результат работы функций *dDFT* и *iDFT*, на примере преобразования изображения 200x248 пикселей (Перед обработкой размер изображения приведен к оптимальному с помощью функции *goToOptimalSize*).

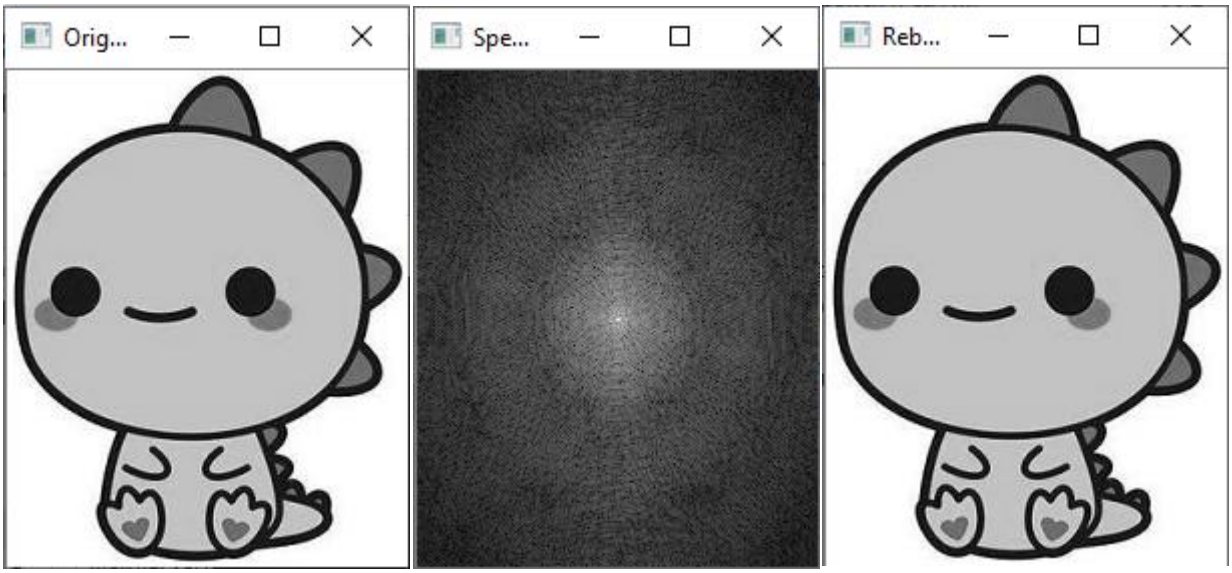


Рисунок 1 – Результат работы функций и спектр изображения

В таблице 1 представлено сравнение работы написанных функций с встроенной функцией opencv.

Таблица 1 – Результаты сравнения

	dDFT	dDFT(opencv)	iDFT	iDFT(opencv)
Время работы, мс	50143	0.54	47764	0.31
Средне-квадратическое отклонение	Re	Im	2.87e-5	
	0.036	0.00054		

*Для изображения, полученного обратным преобразованием, среднеквадратическое отклонение высчитывалось после нормализации.

Время работы функций определялось с помощью библиотеки chrono.

Для получения спектра и фазы используется функция *getSpectreAndPhase*, использующая вещественную и мнимую части образа Фурье, и записывающая результат в вектор изображений (0 элемент – спектр, 1 – фаза). Для вывода

спектра и фазы используются функции *showSpectre* и *showPhase* соответственно, принимающие ранее полученный вектор в качестве аргумента.

2. Фильтр Гаусса низких и высоких частот

Передаточные функции фильтра низких частот (ФНЧ) и фильтра высоких частот (ФВЧ) показаны на формулах 3 и 4 соответственно.

$$H(u,v) = e^{\frac{-D^2(u,v)}{2D_0^2}}, \tag{3}$$

$$H(u,v) = 1 - e^{\frac{-D^2(u,v)}{2D_0^2}}, \tag{4}$$

где $D_0^2 -$ дисперсия; $D(u,v) -$ расстояние от центра фильтра до пикселя с координатами (u,v).

Спектры ФНЧ и ФВЧ до перестановки квадрантов изображены на рисунке 2 слева и справа соответственно.

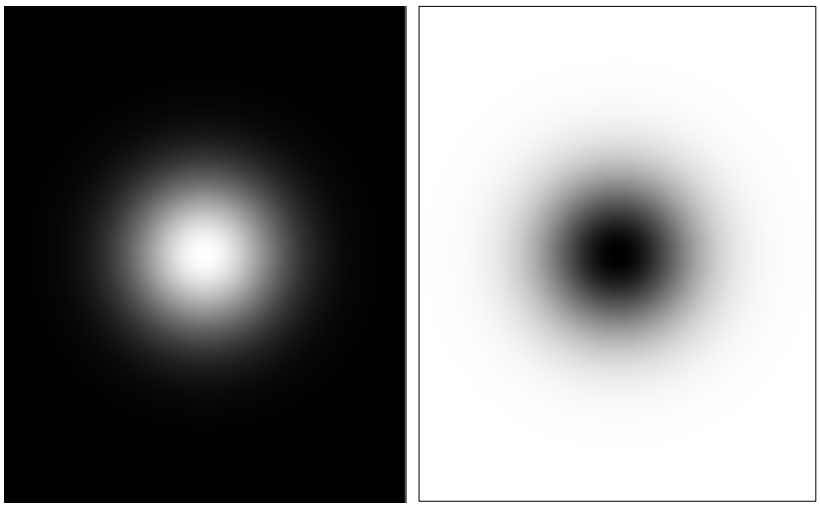


Рисунок 2 – Спектры ФНЧ (слева) и ФВЧ (справа)

Фильтрация ФНЧ и ФВЧ реализована функцией *gaussFilter*, принимающей изображение в формате CV_32FC2 (В одном канале вещественная часть образа, в другом мнимая), дисперсию и флаг (low_pass для ФНЧ и high_pass для ФВЧ). На выходе этой функции получается изображение CV_32FC2 с отфильтрованной вещественной и мнимой частью.

На рисунке 3 представлен результат работы функции *gaussFilter*.

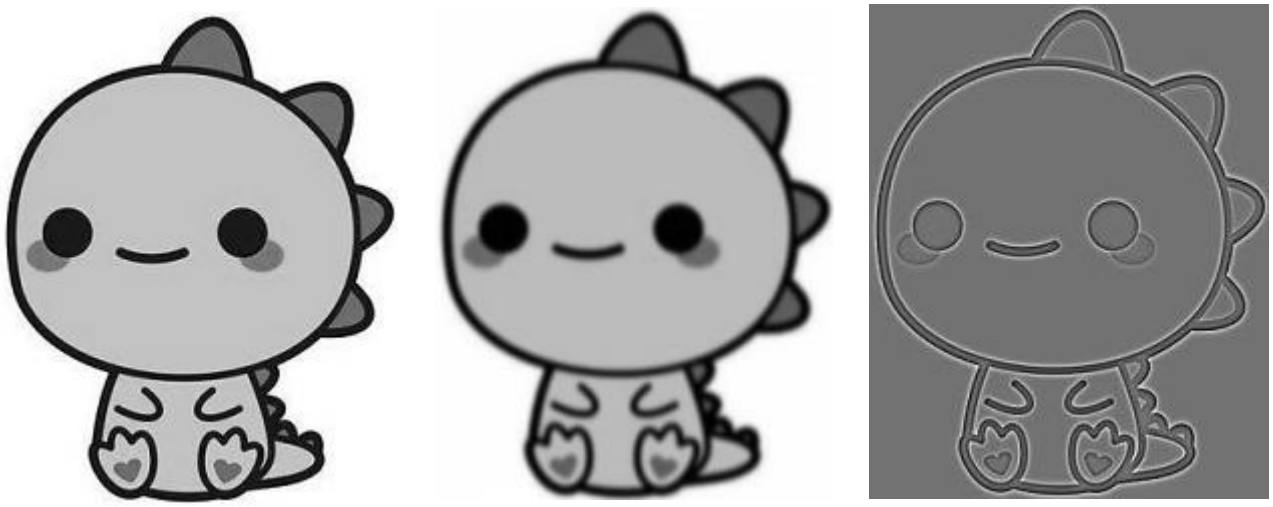


Рисунок 3 – Результат работы ФНЧ (в центре) и ФВЧ (справа)

3. Свертка изображения с ядром фильтра

Для свертки изображения с ядром фильтра используется функция *mulSpec*, принимающая 2 изображения в формате CV_32FC2 и флаг типа bool (если он установлен, то выполнится корреляция вместо свертки). На выходе функции получается изображение типа CV_32FC2.

Для преобразования ядра фильтра в изображение используется функция kernelImg, принимающая ядро фильтра в виде двумерного массива и размер исходного изображения. На выходе получается изображение типа CV_32FC1, которое после прямого преобразования Фурье можно использовать для свертки.

Свертка с ядром фильтра Собеля представлена на рисунках 4 (по горизонтали) и 5 (по вертикали).

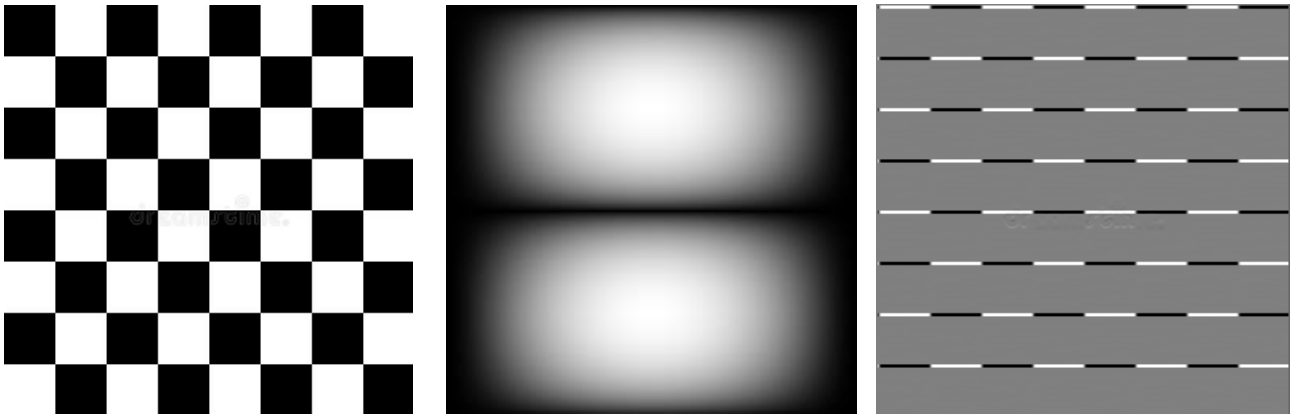


Рисунок 4 – Спектр ядра фильтра Собеля (горизонталь) и результат свертки

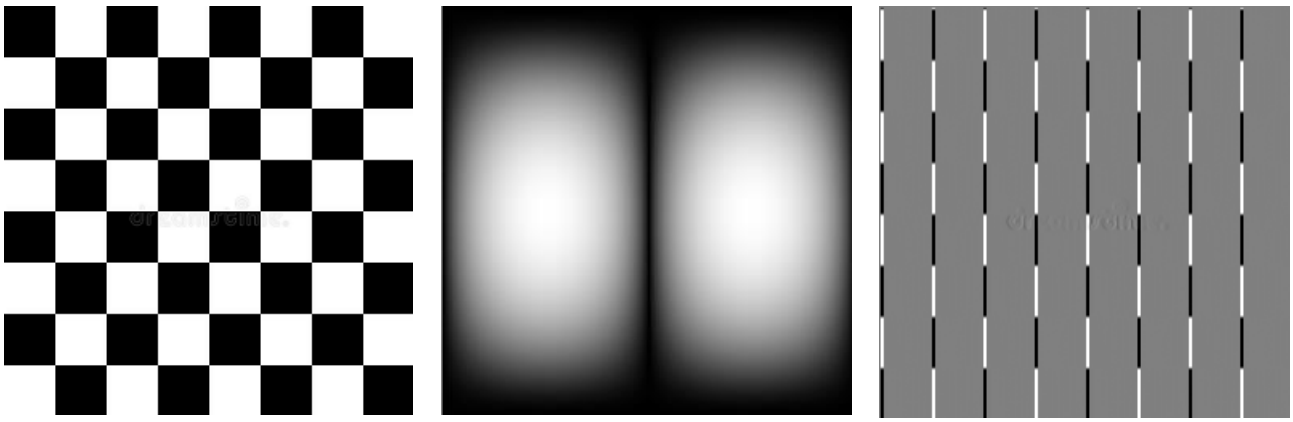


Рисунок 5 – Спектр ядра фильтра Собеля (вертикаль) и результат свертки

На рисунке 6 изображена свертка с ядром усредняющего фильтра.

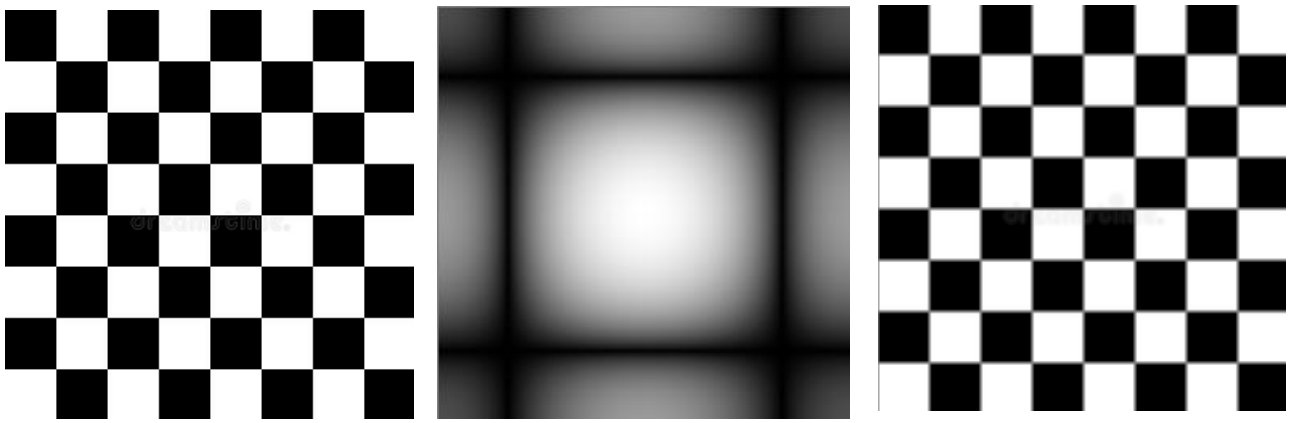


Рисунок 6 – Спектр ядра усредняющего фильтра и результат свертки

Свертка с ядром фильтра Лапласа представлена на рисунке 7.

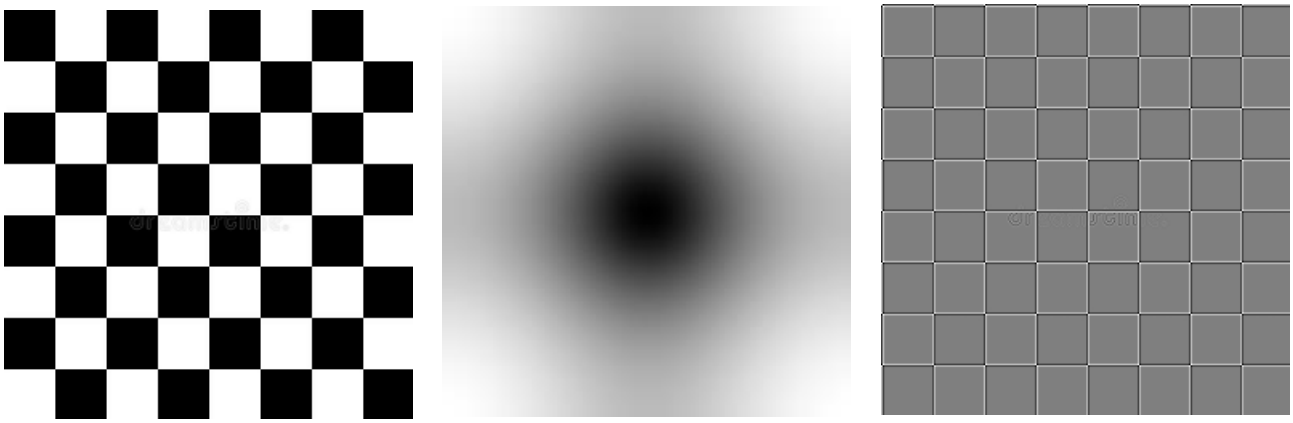


Рисунок 7 – Спектр ядра фильтра Лапласа и результат свертки

4. Корреляция изображений

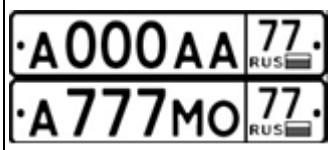
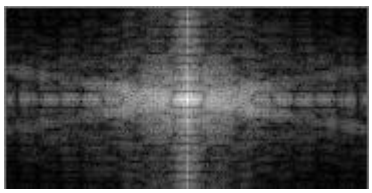
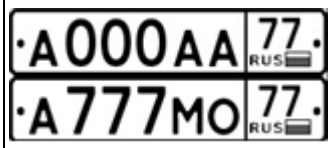
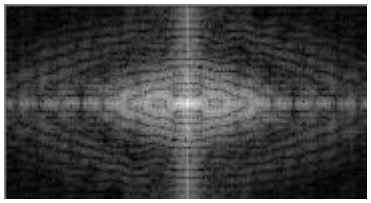
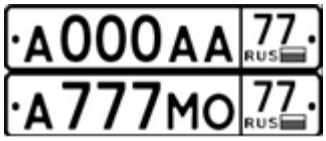
Корреляция изображений выполняется с помощью функции mulSpec, о которой было подробно рассказано в пункте 3. Отличие корреляции от свертки состоит в том, что знак перед мнимой частью одного изображения заменяется на противоположный.

При корреляции изображения должны быть одного размера, поэтому перед прямым преобразованием Фурье изображения приводятся к общему размеру с помощью функции *goToOptimalSize.*

После корреляции выполняется обратное преобразование Фурье. Полученное изображение подвергают пороговой фильтрации.

Результаты корреляции изображения с его частями представлены в таблице 1.

Таблица 1 – Результаты корреляции

Коррелируемые изображения	Итоговый спектр	После обратного преобразования и пороговой фильтрации
 A		
 0		
 7