ЛАБОРАТОРНАЯ РАБОТА №5.
Обработка изображений в частотной области

ЦЕЛЬ РАБОТЫ – Изучение принципов обработки и фильтрации изображений в частотной области; реализация методов частотной обработки изображений на языке Python.

Содержание

1 Теоретические сведения

1.1 Перевод изображения в частотную область

Пусть $f(x, y)$ при $x = 0,1, .. M - 1$ и $y = 0,1, ..., N - 1$, обозначает изображение $M \times N$. Двумерное дискретное преобразование Фурье (ДПФ) изображения f, которое обозначается $F(u, v)$ задается уравнением

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}, \quad (1.1)$$

где $u = 0,1, ... M - 2$ и $v = 0,1, ... N - 1$.

Обратное дискретное преобразование Фурье задается уравнением

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}. \quad (1.2)$$

Таким образом, зная $F(u, v)$ можно всегда восстановить $f(x, y)$ используя (1.2). Величины $F(u, v)$ принято называть *коэффициентами разложения Фурье*.

Значение ДПФ в начале координат частотной области, т.е. $F(0,0)$, называется постоянной составляющей (dc component). Эта терминология пришла из электротехники, где «dc» означает «direct current», постоянный электрический ток (ток с нулевой частотой).

Основной метод визуального анализа ДПФ – это вычисление его ***спектра*** (т.е. модуля комплексных значений $F(u, v)$):

$$|F(u,v)| = \sqrt{\mathrm{Re}^2\{F(u,v)\} + \mathrm{Im}^2\{F(u,v)\}}. \tag{1.3}$$

Фазовый угол $F(u,v)$ задается формулой:

$$\theta(u,v) = \arg F(u,v). \tag{1.4}$$

Напомним, что $\arg z$ – это главное значение аргумента комплексного числа z. Любое комплексное число $z = x + jy$ можно выразить в виде $z = |z|e^{j\arg z}$, где

$$\arg z = \begin{cases} \operatorname{arctg}\left(\dfrac{y}{x}\right), \text{если } x > 0 \\ \pi + \operatorname{arctg}\left(\dfrac{y}{x}\right), \ \text{если } x < 0, \quad y \geq 0, \\ -\pi + \operatorname{arctg}\left(\dfrac{y}{x}\right), \ \text{если } x < 0, \quad y < 0, \\ \dfrac{\pi}{2}, \quad \text{если } x = 0, \quad y > 0, \\ -\dfrac{\pi}{2}, \quad \text{если } x = 0, \quad y < 0. \end{cases} \tag{1.5}$$

Ниже приведен пример вычисления на языке Python модуля и аргумента комплексного числа $z = \frac{\sqrt{2}}{2} + \frac{j\sqrt{2}}{2}$. Для вычисления модуля и аргумента используется функции `abs` и `angle` библиотеки `numpy`.

```
>>> z = 1/np.sqrt(2) + 1j/np.sqrt(2)
>>> abs_z = np.abs(z)
>>> arg_z = np.angle(z)
>>> print(f'Abs. val. = %1.5f' % abs_z)
>>> print(f'Argument  = %1.5f' % arg_z)

Abs. val. = 1.00000
Argument  = 0.78540
```

1.1.1 Свойства двумерного ДПФ

Поскольку изображение $f(x,y)$ вещественно, то его ДПФ является сопряженно симметричным начала координат, т.е.

$$F(u,v) = F^*(-u,-v), \tag{1.6}$$

что означает симметрию спектра Фурье относительно начала координат:

$$|F(u,v)| = |F^*(-u,-v)|. \tag{1.7}$$

Сделав прямую подстановку в уравнения для $F(u,v)$, можно показать, что

$$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N). \tag{1.8}$$

Т.е. ДПФ $F(u,v)$ является периодической по обеим переменным u и v, перерды которой равны M и N соответственно.

1.1.2 Центрирование ДПФ

На практике очень часто удобно смещать начало координат частотной области в центр массива. Эта процедура называется ***центрированием***.

Чтобы понять, осуществляется центрирование рассмотрим еще раз определение двумерного ДПФ (1.1). Теперь давайте сдвинем частотные координаты следующим образом $u' = u - (M/2)$ и $v' = v - (N/2)$, теперь точка $(u', v') = (0{,}0)$ находится в центре массива. Центрированное ДПФ $F(u', v')$ определяется как

$$F(u',v') = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-2\pi j\left(\frac{\left(u-\frac{M}{2}\right)x}{M}+\frac{\left(u-\frac{N}{2}\right)y}{N}\right)} \qquad (1.9)$$

Преобразовывая данное выражение и отделяя части, которые не зависят от u и v, и используя тождество $e^{j\pi} = -1$ приходим к выражению

$$F(u',v') = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}(-1)^{(x+y)} f(x,y)e^{-2\pi j\left(\frac{ux}{M}+\frac{uy}{N}\right)}. \qquad (1.10)$$

Таким образом, центрированное ДПФ является обычным ДПФ от измененного изображения $(-1)^{(x+y)}f(x,y)$. Таким образом есть простой способ вычислить центрированное ДПФ:

$$F(u',v') = F\left(u - \frac{M}{2}, v - \frac{N}{2}\right) = \mathrm{DFT}_{2D}\{(-1)^{(x+y)}f(x,y)\}, \qquad (1.11)$$

где $\mathrm{DFT}_{2D}\{\cdot\}$ –оператор вычисления двумерного ДПФ по выражению (1.1).

Есть ещё один способ получения центрированного ДПФ. Допустим, что у нас имеется вычисленное ДПФ $F(u,v)$ от изображения $f(x,y)$. Можно показать, что центрированное ДПФ определяется путем диагональной перестановки квадрантов ДПФ, как показано на рис. 1.

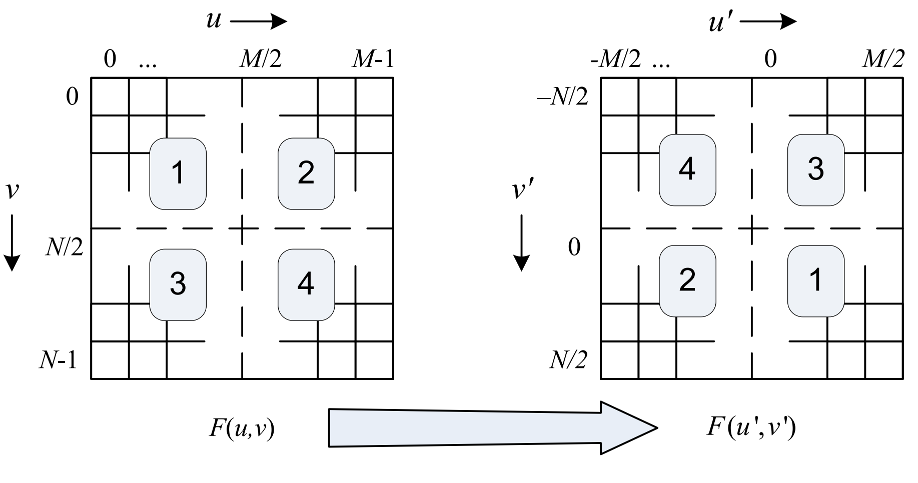

Рисунок 1 – Вычисление центрированного ДПФ из нецентрированного путем разбиения массива на четыре квадранта и их диагональной перестановки

Операцию такой перестановки данных в массиве реализует функция из Python `numpy.fft.fftshift()`.

1.1.3 Логарифмическое преобразование спектра ДПФ

Логарифмическое преобразование состоит в сжатии динамического диапазона. Спектр Фурье часто имеет диапазон значений от 0 до 10^6 и выше. Если линейно масштабировать этот диапазон в интервал с 8-ми битной градацией, то при визуализации наиболее яркие пиксели будут доминировать, что приведет к утрате менее ярких участков спектра. Если применить функцию ln(), то динамический диапазон с амплитудой, например, 10^6, сократится до 14, что намного удобнее.

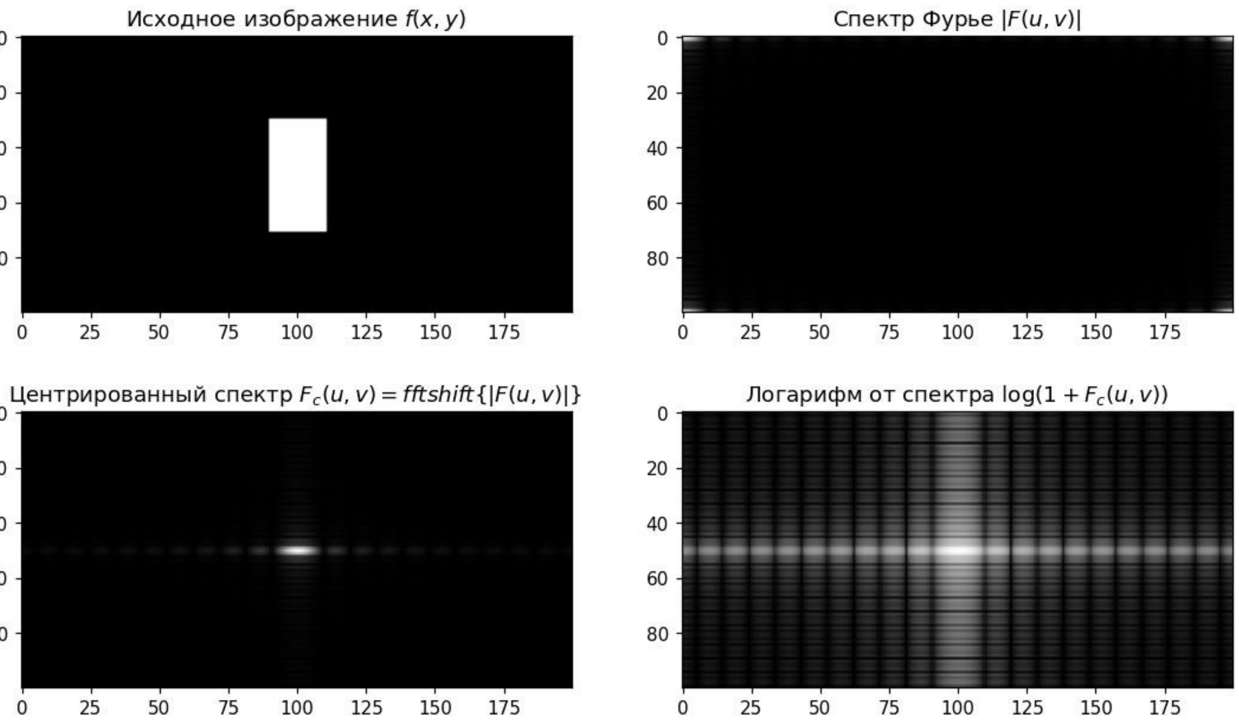

Рисунок 2 – Вычисление центрированного ДПФ из нецентрированного путем разбиения массива на четыре квадранта и их диагональной перестановки

Для выполнения логарифмического преобразования можно использовать формулу

$$c\ln(F(u,v)+1), \tag{1.12}$$

где c – некоторая константа.

На рис. 2 показано отличие обычного спектра от логарифмического. Визуальное улучшение по сравнению с исходным не вызывает сомнения.

1.2 Основные шаги фильтрации в частотной области

Линейная фильтрация, выполняемая в пространственной области, может быть выполнена и в частотной. Основанием для этого служит теорема о свертке

$$f(x,y) * h(x,y) \overset{\mathrm{DFT}_{2D}}{\longleftrightarrow} H(u,v)F(u,v). \tag{1.13}$$

Здесь символ «∗» означает операцию свертки двух функций, а выражение по обе стороны от двусторонней стрелки соответствующие пары преобразований Фурье. Т.е. фильтрация изображения $f(x,y)$ в пространственной области фильтром $h(x,y)$ выполняется при помощи операции свертки. Этого же результата можно достичь в частотной области умножив $F(u,v)$ на $H(u,v)$, где $F(u,v) = \mathrm{DFT}_{2D}\{f(x,y)\}$, а $H(u,v) = \mathrm{DFT}_{2D}\{h(x,y)\}$.

Предполагается, что передаточная функция фильтра $H(u,v)$ имеет те же размеры, что и исходное изображение.

При фильтрации изображения в частотной области обычно выполняют следующие шаги:

1) расширение исходного изображения `f` размера $M \times N$ до размеров $P \times Q$ – `f_ext`;

2) вычисление ДПФ `F` от расширенного изображения `f_ext`;

3) формирование передаточной функции фильтра `H` размера $P \times Q$ одним из описываемых далее методов;

4) Умножение ДПФ изображение на передаточную функцию: `G=H×F`;

5) Вычисление вещественной части обратного ДПФ от `G`: `g=real(ifft2(G))`;

6) Вырезание верхнего левого прямоугольника из `g` исходных размеров $M \times N$.

1.3 Расширение изображения

Есть несколько основных типов расширения изображения

– константное расширение (0 или 255)

– зеркальное отражение («reflect»)

– симметрическое расширение («symmetric»)

– периодическое повторение («wrap»)

Для выполнения этого действия необходимо использовать функцию `numpy.pad()`. Ниже приведен пример расширения изображения.

```
[1]: baboon_gray_img = plt.imread("img\\baboon.tiff")
[2]: img_const = np.pad(baboon_gray_img,(100,200),'constant', con-
stant_values=(255, 255))
[3]: img_sym  = np.pad(baboon_gray_img,(256,256),'symmetric')
[4]: img_ref  = np.pad(baboon_gray_img,(256,256),'reflect')
[5]: img_wrap = np.pad(baboon_gray_img,(256,256),'wrap')
```

Результат выполнения данных команд показан на рис. 3.

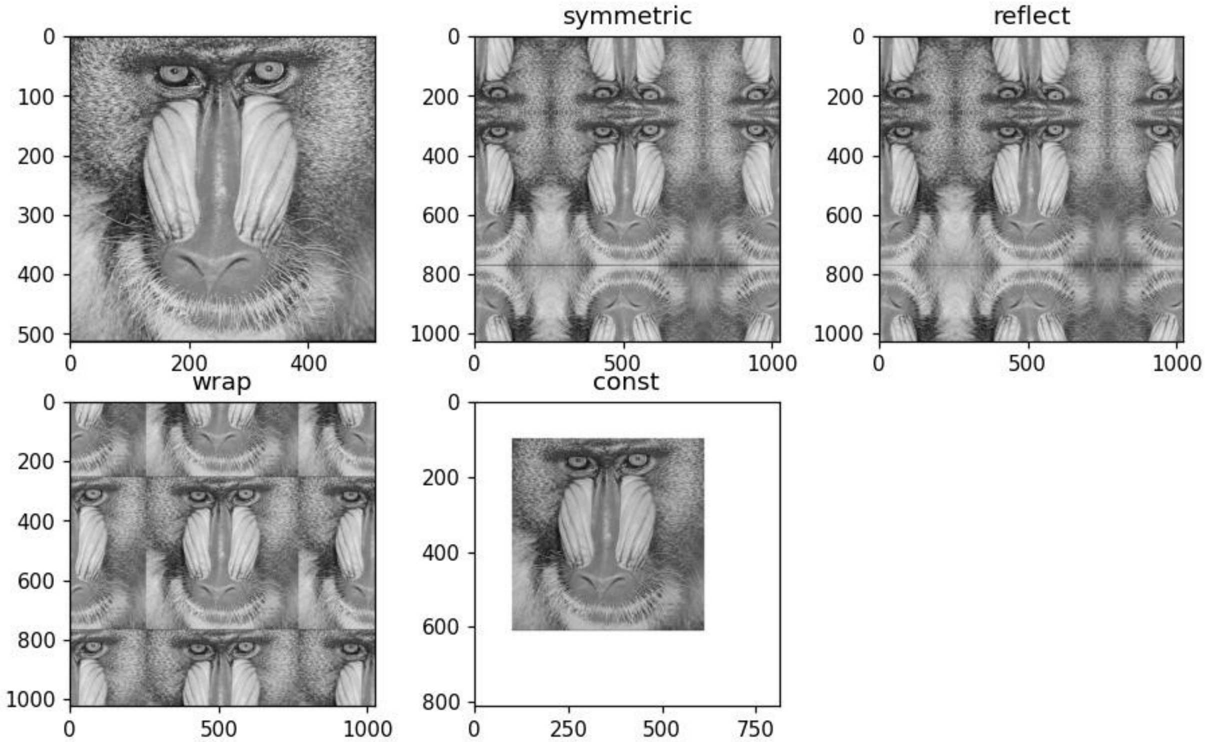

Рисунок 3 – Расширение изображения

1.4 Низкочастотные фильтры

Идеальный низкочастотный фильтр (ILPF, Ideal Lowpass Filter) имеет передаточную функцию

$$H(u,v) = \begin{cases} 1 \text{ при } D(u,v) \le D_0 \\ 0 \text{ при } D(u,v) > D_0, \end{cases} \tag{1.14}$$

где D_0 – неотрицательное число, а $D(u,v)$ – расстояние от центра фильтра до точки (u,v). Геометрическое место точек (u,v) для которых $D(u,v) = D_0$ является окружностью.

Низкочастотный фильтр Баттерворта (BLPF, Butterworth LowPass Filter) порядка n с обрезкой частот на расстоянии D_0 от начала координат имеет вид

$$H(u,v) = \frac{1}{1 + \left(\frac{D(u,v)}{D_0}\right)^{2n}} \tag{1.15}$$

Передаточная функция гауссова низкочастотного фильтра (GLPF, Gaussian LowPass Filter) задается выражением

$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}, \tag{1.16}$$

где σ – это стандартное отклонение. Если установить $\sigma = D_0$, то получится следующее выражение:

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}. \tag{1.17}$$

1.5 Повышение резкости при частотной фильтрации

Высокочастотная фильтрация повышает резкость изображения, ослабляя низкие частоты и оставляя высокие. Имея передаточную функцию $H_{lp}(u,v)$ низкочастотного фильтра, можно получить передаточную функцию соответствующего высокочастотного фильтра с помощью формулы

$$H_{hp}(u,v) = 1 - H_{lp}(u,v). \tag{1.18}$$

1.6 Фильтрация с усилением высоких частот

ВЧ фильтры отбрасывают постоянную составляющую так, что в итоге средняя интенсивность изображения становится равной нулю. Чтобы это компенсировать нужно добавить смещение к ВЧ фильтру. Если смещение комбинируется вместе с умножением передаточной функции на коэффициент усиления, то итоговая процедура будет называться фильтрацией с усилением высоких частот:

$$H_{hfe}(u,v) = a + bH_{hp}(u,v), \tag{1.19}$$

где a – смещение, b – коэффициент усиления.

2 Порядок выполнения работы

1) Написать функцию на языке Python, которая вычисляет двумерное преобразование Фурье, с использованием функции вычисления одномерного ДПФ `numpy.fft.fft()`. Загрузите монохромное изображение размером не более 1000x1000 пикселей и вычислите его ДПФ-образ. Отобразите на экране модуль ρ и фазу θ ДПФ-коэффициентов:

$$X(u,v) = \rho(u,v)e^{j\theta(u,v)}.$$

2) Написать функцию на языке Python, которая вычисляет обратное двумерное преобразование Фурье без использования функций из сторонних библиотек. Проверьте: можете ли вы точно восстановить изображение по его Фурье-образу, полученному в пункте 1.

3) Исследуйте влияние амплитуды и фазы коэффициентов Фурье-образа на изображения:

а) установите фазу $\theta(u,v)$ в нуль и восстановите изображение, отобразите результат на экране;

б) установите модуль $\rho(u,v)$ в 1 и восстановите изображение, результат отобразите на экране. Опишите полученный эффект в обоих случаях.

4) Выполните фильтрацию изображений в Фурье-области, используя представление в полярной системе координат (рис. 4):

а) установите в нуль Фурье-коэффициенты, чьи частоты располагаются на расстоянии $K_1 = 3n_0$ от начала координат (n_0 – номер вашего варианта); восстановите изображение, какой эффект вы наблюдаете? Опишите признаки, которые пропали в результате операции; Наблюдаете ли вы какие-то искажения?

б) Предыдущее преобразование соответствует фильтру нижних частот; в данном задании построим полосовой фильтр. Для этого установите все коэффициенты, которые не попадают в кольцо $K_1 \leq \sqrt{u^2 + v^2} \leq K_2$, в нуль (табл. 1); восстановите изображение, какой эффект вы наблюдаете? Опишите признаки, которые пропали в результате операции.

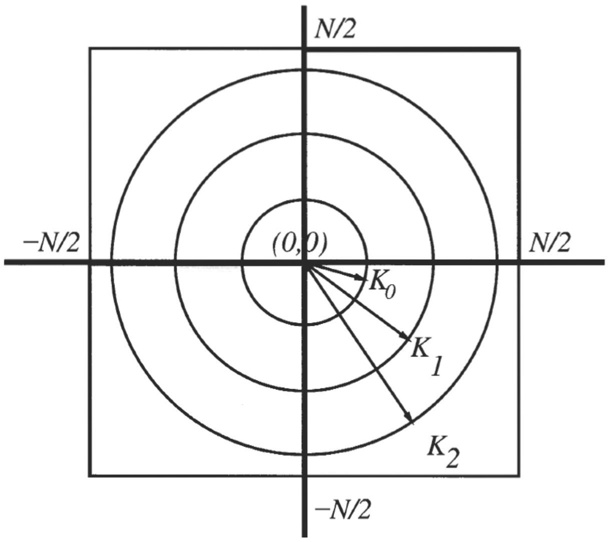

Рисунок 4 – Двумерная Фурье-область изображения

Таблица 1 – Варианты для задания 4 ($t = \min(w, h)$, где w – ширина изображения, h – высота изображения)

Номер варианта (n_0)	K_1	K_2
1	$n_0 + 50$	$\lfloor t/2.2 \rfloor$
2		$\lceil t/3.1 \rceil$
3	$8n_0$	$\lfloor t/2.5 \rfloor$
4		$\left\lceil te^{-\frac{0.7t}{t-120}} \right\rceil$
5	$\mathrm{round}(e^{n_0/\sqrt{2}})$	$\lfloor t/2 - 4\ln(n_0) \rfloor$
6		
7	$\left\lceil -20\ln\left(\frac{1}{n_0}\right) \right\rceil$	$\lceil t - 3n_0 \rceil$
8		

5) Постройте передаточную функцию фильтра высоких частот (табл. 2).

Таблица 2 – Варианты для задания 2

Номер варианта	Значение параметра D_0	Тип НЧ фильтра, используемого для построения фильтра ВЧ
1	100	Идеальный низкочастотный фильтр
2	152	Низкочастотный фильтр Баттерворта ($n = 2$)
3	120	Гауссов низкочастотный фильтр ($\sigma = D_0$)

4	200	Идеальный низкочастотный фильтр
5	169	Низкочастотный фильтр Баттерворта ($n = 3$)
6	92	Гауссов низкочастотный фильтр ($\sigma = 1.1 \cdot D_0$)
7	182	Низкочастотный фильтр Баттерворта ($n = 10$)
8	155	Гауссов низкочастотный фильтр ($\sigma = 1.4 \cdot D_0$)

6) Выполните высокочастотную фильтрацию изображения с использованием фильтра из задания 5. После чего примените к изображению эквализацию гистограммы. Является ли данная процедура эффективным методом повышения резкости и улучшения контраста?

а) Подумайте, влияет ли порядок процедур на окончательный результат.

б) Если порядок процедур важен, то дайте обоснованый ответ, какую процедуру использовать вначале.

7) Выполните фильтрацию с усилением высоких частот, используя формулу (1.19). Подберите константы a и b для получения наилучшего эффекта.

8) Оформить отчет в соответствии с СТП 01-2017.

3 Литература

1. Гонсалес, Р., Вудс, Р., Эддинс, С. *Цифровая обработка изображений в среде MATLAB* – М.: Техносфера, 2006. – 616 с.