



## Математические методы обработки изображений.

В настоящем пособии излагаются методы, которые используются при обработке изображений. Перед вами часть, которая знакомит с методами обработки изображений в частотной области.

### Методы обработки изображений в частотной области.

#### Оглавление

4. Преобразование Фурье .....	1
4.1 Ряды Фурье и одномерное дискретное преобразование Фурье .....	1
4.2 Двумерное дискретное преобразование Фурье .....	10
4.3 Вычисление и визуализация дискретного преобразования Фурье .....	12
4.4 Фильтрация в частотной области.....	19

## 4. Преобразование Фурье

### 4.1 Ряды Фурье и одномерное дискретное преобразование Фурье

Периодические функции с периодом  $2 \cdot L$  можно разложить в ряд по синусам и косинусам.

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [a_n \cos(n\pi x/L) + b_n \sin(n\pi x/L)], \quad (1)$$

в котором коэффициенты Фурье  $a_n$  и  $b_n$  находятся по формулам

$$a_n = \frac{1}{L} \int_{-L}^L f(x) \cos(n\pi x/L) dx, \quad n = 0, 1, 2, \dots, \quad (2)$$
$$b_n = \frac{1}{L} \int_{-L}^L f(x) \sin(n\pi x/L) dx, \quad n = 1, 2, \dots$$

В ряд Фурье можно разложить только периодическую функцию. Но если функция определена на конечном интервале, например на интервале  $-L \leq x \leq L$ , то ее всегда можно периодически с периодом  $2 \cdot L$  продолжить на всю числовую ось и затем использовать разложение для периодической функции.

Ряды Фурье удобно записывать в комплексном виде. Поскольку

$$\begin{aligned}\cos(n\pi x/L) &= \frac{e^{in\pi x/L} + e^{-in\pi x/L}}{2} \\ \sin(n\pi x/L) &= \frac{e^{in\pi x/L} - e^{-in\pi x/L}}{2i} = \frac{i}{2}(e^{-in\pi x/L} - e^{in\pi x/L})\end{aligned}\quad (3)$$

то, подставив (3) в (1), получим

$$\begin{aligned}f(x) &= \frac{a_0}{2} + \sum_{n=1}^{\infty} \left\{ \frac{1}{2} a_n (e^{in\pi x/L} + e^{-in\pi x/L}) + b_n \frac{i}{2} (e^{-in\pi x/L} - e^{in\pi x/L}) \right\} = \\ &= \frac{a_0}{2} + \sum_{n=1}^{\infty} \left\{ \frac{1}{2} (a_n - ib_n) e^{in\pi x/L} + \frac{1}{2} (a_n + ib_n) e^{-in\pi x/L} \right\}\end{aligned}$$

Полагая

$$c_0 = \frac{a_0}{2}, \quad c_n = \frac{1}{2}(a_n - ib_n), \quad c_{-n} = \frac{1}{2}(a_n + ib_n)$$

имеем

$$f(x) = \sum_{n=-\infty}^{\infty} c_n e^{in\pi x/L} \quad (4)$$

где, очевидно  $c_{-n} = \overline{c_n}$ ,  $n = 1, 2, \dots$ . Вспоминая, что  $\cos \alpha \pm i \sin \alpha = e^{\pm i \alpha}$ , будем иметь

$$\begin{aligned}c_n &= \frac{1}{2}(a_n - ib_n) = \frac{1}{2L} \int_{-L}^L f(x) (\cos(n\pi x/L) - i \sin(n\pi x/L)) dx = \frac{1}{2L} \int_{-L}^L f(x) e^{-in\pi x/L} dx \\ c_{-n} &= \frac{1}{2}(a_n + ib_n) = \frac{1}{2L} \int_{-L}^L f(x) (\cos(n\pi x/L) + i \sin(n\pi x/L)) dx = \frac{1}{2L} \int_{-L}^L f(x) e^{in\pi x/L} dx\end{aligned}$$

или, объединив обе формулы и добавив случай  $n=0$ ,

$$c_n = \frac{1}{2L} \int_{-L}^L f(x) e^{-in\pi x/L} dx, \quad n = 0, \pm 1, \pm 2, \dots \quad (5)$$

Подставив (5) в (4), получим

$$f(x) = \frac{1}{2L} \sum_{n=-\infty}^{\infty} \left( \int_{-L}^L f(x) e^{-in\pi x/L} dx \right) e^{in\pi x/L} \quad (6)$$

Итак, для периодической с периодом  $2L$  функции мы записали ряд Фурье в комплексной форме, и нашли выражения для его коэффициентов.

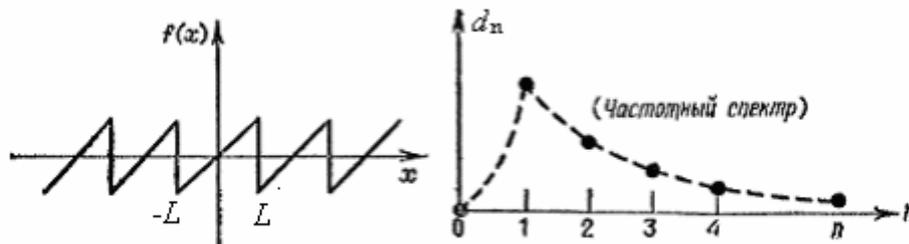
Разложение в ряд Фурье можно интерпретировать как представление функции  $f(x)$  последовательностью чисел  $d_n$  вида

$$d_n = \sqrt{a_n^2 + b_n^2}, \quad n = 0, 1, 2, \dots,$$

где коэффициенты  $d_n$  служат мерой вклада соответствующих частотных компонент в функцию  $f(x)$ . Например, пилообразная функция  $f(x)$  разлагается в ряд Фурье

$$f(x) = \frac{2L}{\pi} \left[ \sin(\pi x/L) - \frac{1}{2} \sin(2\pi x/L) + \frac{1}{3} \sin(3\pi x/L) - \dots \right] \quad (7)$$

и, следовательно, частотный спектр  $\{d_n\}$  записывается в виде  $d_n = 2L/n\pi$  для  $n=1,2,\dots$  и  $d_0 = |a_0| = 0$ .



Пилообразная функция и ее частотный спектр  $d_n$

Действительно

$$a_n = \frac{1}{L} \int_{-L}^L x \cos(n\pi x/L) dx = 0, \quad n=0, 1, 2, \dots,$$

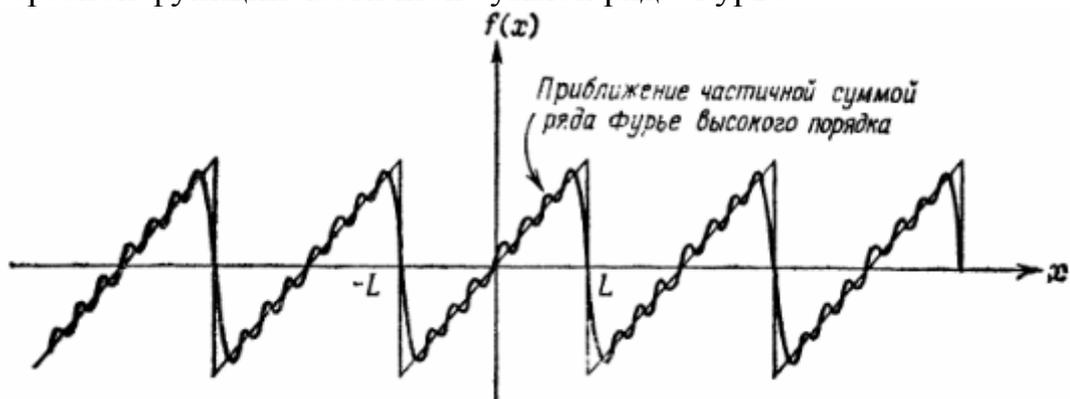
$$b_n = \frac{1}{L} \int_{-L}^L x \sin(n\pi x/L) dx = -(-1)^n (2L/n\pi), \quad n=1, 2, \dots$$

и, следовательно,  $d_n = 2L/n\pi$  для  $n=1,2,\dots$  и  $d_0 = |a_0| = 0$ .

Если использовать конечное число коэффициентов последовательностей  $\{a_n\}$  и  $\{b_n\}$ , то частичная сумма вида

$$f(x) = \frac{a_0}{2} + \sum_{n=1}^N [a_n \cos(n\pi x/L) + b_n \sin(n\pi x/L)],$$

не будет совпадать с исходной функцией, однако, чем больше  $N$ , тем точнее приближение. Например, следующий рисунок иллюстрирует приближение пилообразной функции частичной суммой ряда Фурье



**Пример 1.** Построения график частичной суммы в Matlab. Для этого создадим функцию  $f(x)$ , вычисляемую по формуле (7)

```
function z=furierser1(x,L,Nmax)
% вычисление Nmax членов ряда фурье периодической
% пилообразной функции y=x на отрезке [-L L]
% аргумент x должен быть вектором, на котором следует
% вычислять значение ряда
z=[];
for t=x
    s=0;
    for k=1:Nmax
```

```

s=s+((-1)^(k+1))*2*L/(k*pi)*sin(k*pi*t/L);
end
z=[z s];
end

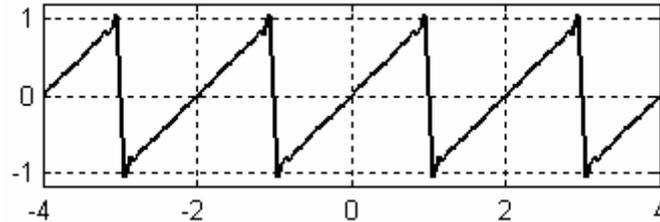
```

Теперь для построения графика частичной суммы выполним команды (соберите команды в сценарий)

```

x=-4:0.05:4; z=furierser1(x,1,20);
plot(x,z,'k','LineWidth',2); grid on;
axis equal; axis([-4 4 -1.2 1.2]);

```



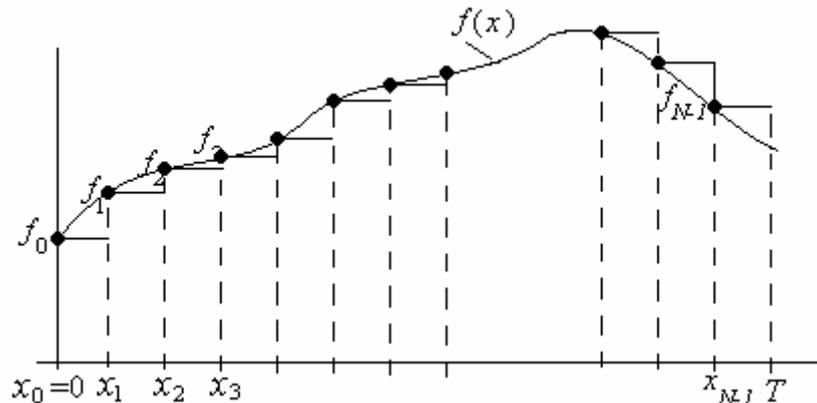
Увеличивайте Nmax и смотрите, что происходит.

Фактически, две последовательности  $\{a_n\}$  и  $\{b_n\}$  содержат всю информацию о функции. Чем больше элементов этих последовательностей  $\{a_n\}$  и  $\{b_n\}$  мы знаем, тем точнее мы можем восстановить функцию.

Рассмотрим теперь некоторую периодическую функцию  $f(x)$  с периодом  $T=2\cdot L$ . Разложим ее в ряд Фурье

$$f(x) = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{\pi k x}{L}} \quad (8)$$

Проведем дискретизацию функции так, чтобы на периоде  $T$  было  $N$  отсчетов.



Т.е. заменим функцию  $f(x)$  на кусочно-постоянную функцию со значениями  $f_n = f(x_n)$  на участках  $[x_n, x_{n+1})$ , где  $x_n = \frac{n}{N}T = \frac{n}{N}2L$ . Точки  $(x_n, f_n)$  принадлежат графику функции  $f(x)$  и для них выполняется равенство (8). Поэтому

$$f_n = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{\pi k x_n}{L}} = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{\pi k}{L} \cdot \frac{2nL}{N}} = \sum_{k=-\infty}^{\infty} c_k e^{i \frac{2\pi}{N} \cdot k n} = \dots +$$



Если обозначить  $\alpha = e^{i \frac{2\pi}{N} \cdot (k-m)}$ , то внутренняя сумма в последнем выражении при  $\alpha \neq 1$  т.е.  $k \neq m$  будет представлять сумму геометрической прогрессии

$$\sum_{n=0}^{N-1} \alpha^n = \frac{1 - \alpha^N}{1 - \alpha} = \frac{1 - e^{i \frac{2\pi}{N} \cdot (k-m)N}}{1 - e^{i \frac{2\pi}{N} \cdot (k-m)}} = \frac{1 - e^{i 2\pi \cdot (k-m)}}{1 - e^{i \frac{2\pi}{N} \cdot (k-m)}} = 0, k \neq m$$

А при  $\alpha = 1$ , т.е. при  $k = m$ , мы получим  $\sum_{n=0}^{N-1} 1 = N$ . Т.е.

$$\sum_{n=0}^{N-1} e^{i \frac{2\pi}{N} \cdot (k-m)n} = N \cdot \delta_{km}, \quad (10)$$

где  $\delta_{km} = \begin{cases} 1, & k = m \\ 0, & k \neq m \end{cases}$  - символ Кронекера. В результате получаем

$$\sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} \cdot mn} = N \cdot \sum_{k=0}^{N-1} F_k \cdot \delta_{km} = N \cdot F_m$$

Т.о. для коэффициентов  $F_k$  мы получили представление

$$F_k = \frac{1}{N} \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi k}{N} \cdot n}. \quad (11)$$

Т.о. формулы прямого (10)(11) и обратного (9) дискретного преобразования Фурье имеют следующий вид

$$F_k = \sum_{n=0}^{N-1} f_n e^{-i \frac{2\pi}{N} \cdot kn}, \quad f_n = \frac{1}{N} \sum_{k=0}^{N-1} F_k e^{i \frac{2\pi}{N} \cdot kn} \quad (12)$$

Эти преобразования составляют пару преобразований Фурье, а входящие в них функции образуют Фурье-пару. Часто это обозначают следующим образом  $f(x) \Leftrightarrow F(u)$ .

Перейдем от индексного обозначения к функциональному. Преобразование Фурье дискретной функции  $f(x)$  одной переменной,  $x=0, 1, 2, \dots, M-1$ , задается равенством

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-i \frac{2\pi ux}{M}}, \quad u = 0, 1, 2, \dots, M-1 \quad (13)$$

По заданному Фурье – преобразованию можно восстановить исходную функцию при помощи обратного Дискретного Преобразования Фурье (ДПФ):

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{i \frac{2\pi ux}{M}}, \quad x = 0, 1, 2, \dots, M-1 \quad (14)$$

Множитель  $1/M$  иногда ставится в формуле, определяющей обратное, а не прямое преобразование Фурье. Реже оба равенства содержат множитель  $1/\sqrt{M}$ . Положение множителя не имеет значения. Единственное требование при использовании двух множителей состоит в том, чтобы их произведение равнялось  $1/M$ .

Как и исходная функция  $f(x)$ , Фурье-образ  $F(u)$  является дискретной величиной и содержит то же число элементов.

Покажем, что формулы (13) и (14) образуют Фурье – пару. Равенство (10) выражает свойство ортогональности экспонент

$$\sum_{x=0}^{M-1} e^{i \frac{2\pi r x}{M}} e^{-i \frac{2\pi u x}{M}} = \sum_{x=0}^{M-1} e^{i \frac{2\pi}{M} \cdot (r-u)x} = M \cdot \delta_{ru}$$

Тогда, подставляя  $f(x)$  в виде (14) в правую часть формулы (13), получаем

$$\begin{aligned} \frac{1}{M} \sum_{x=0}^{M-1} \left( \sum_{r=0}^{M-1} F(r) e^{i \frac{2\pi r x}{M}} \right) e^{-i \frac{2\pi u x}{M}} &= \frac{1}{M} \sum_{r=0}^{M-1} F(r) \left( \sum_{x=0}^{M-1} e^{i \frac{2\pi r x}{M}} e^{-i \frac{2\pi u x}{M}} \right) = \\ &= \frac{1}{M} \sum_{r=0}^{M-1} F(r) \cdot M \cdot \delta_{ru} = F(u), \end{aligned}$$

т.е.

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} \left( \sum_{r=0}^{M-1} F(r) e^{i \frac{2\pi r x}{M}} \right) e^{-i \frac{2\pi u x}{M}} \quad (15)$$

Аналогично, подставляя  $F(u)$  в виде (13) в правую часть формулы (14), получаем

$$\begin{aligned} \sum_{u=0}^{M-1} \left( \frac{1}{M} \sum_{r=0}^{M-1} f(r) e^{-i \frac{2\pi u r}{M}} \right) e^{i \frac{2\pi u x}{M}} &= \frac{1}{M} \sum_{r=0}^{M-1} f(r) \sum_{u=0}^{M-1} \left( e^{-i \frac{2\pi u r}{M}} e^{i \frac{2\pi u x}{M}} \right) = \\ &= \frac{1}{M} \sum_{r=0}^{M-1} f(r) \cdot M \cdot \delta_{rx} = f(x), \end{aligned}$$

т.е.

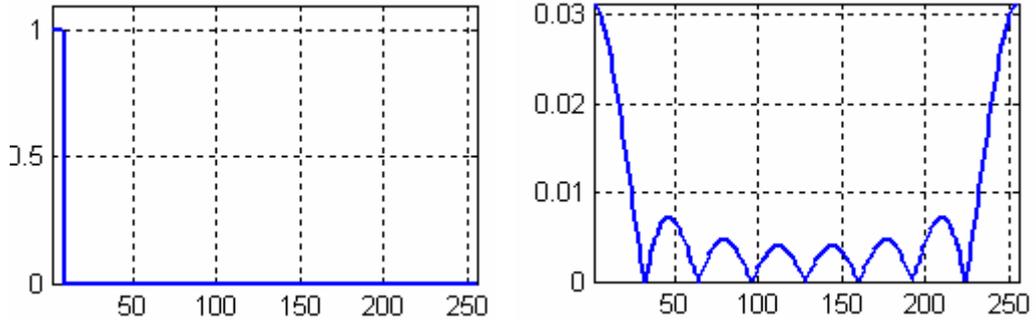
$$f(x) = \sum_{u=0}^{M-1} \left( \frac{1}{M} \sum_{r=0}^{M-1} f(r) e^{-i \frac{2\pi u r}{M}} \right) e^{i \frac{2\pi u x}{M}} \quad (16)$$

Равенства (15) и (16) показывают, что формулы (13) и (14) образуют Фурье – пару.

Область значений переменной  $u$ , на которой принимает свои значения функция  $F(u)$  называется частотной областью. Каждый из  $M$  элементов функции  $F(u)$  называется частотной компонентой.

Даже если функция  $f(x)$  вещественная, ее преобразование Фурье является, как правило, комплексным. Для визуального анализа преобразования Фурье используется спектр Фурье, т. е. абсолютная величина функции  $F(u)$ . Если  $F(u) = R(u) + i \cdot I(u)$ , то спектр Фурье задается выражением  $|F(u)| = \sqrt{R^2(u) + I^2(u)}$ .

**Пример 2.** Фурье спектр простой одномерной функции. Рассмотрим функцию  $f(x)$  и ее спектр Фурье, графики которых показаны на следующем рисунке



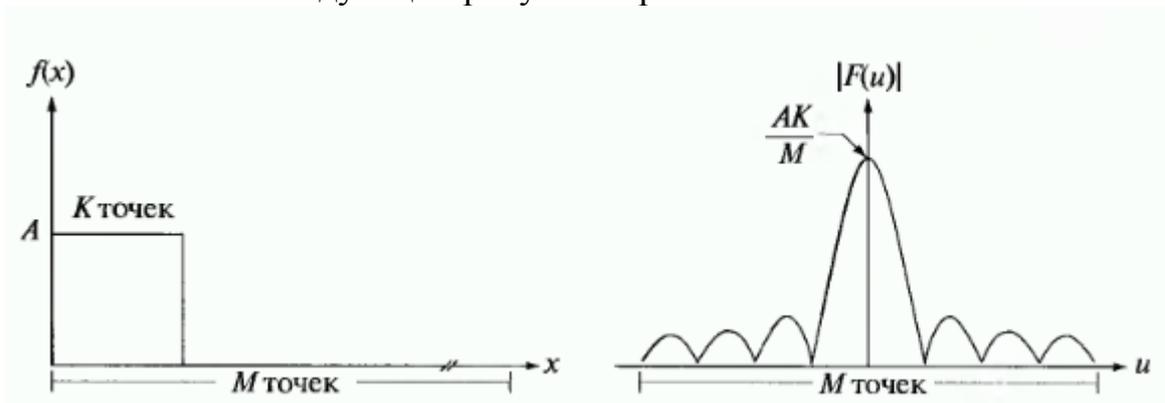
Функция  $f(x)$  равна 1 на узком участке в окрестности 0 и равна 0 в остальных точках. И функция  $f(x)$  и ее Фурье – образ  $F(u)$  являются дискретными, однако на графиках точки соединены между собой для лучшего зрительного восприятия. Эти графики построены с использованием следующих команд (ex28.m)

```
M=256;
x=1:M; u=1:M;
fx=zeros(1,M); % fx(1:M/2)=1;
fx(1:M/32)=1;
subplot(1,2,1); plot(x,fx,'LineWidth',2); % исходная функция
axis([1 M 0 max(abs(fx*1.1))]);
grid on;

fu=(fx*exp(-i*2*pi*u'*x/M))/M;
subplot(1,2,2);plot(u,abs(fu),'LineWidth',2); % ПФ
axis([1 M 0 max(abs(fu))]);
grid on;
set(gcf,'Color',[1 1 1]);
```

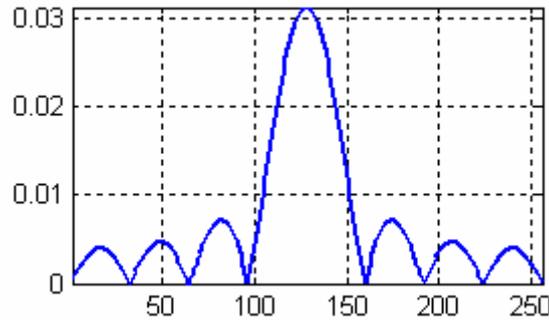
Параметры исходной функции – ступеньки пояснены на следующем рисунке слева. В рассматриваемом примере  $M=256$ ,  $A=1$  и  $K=8$  точек.

Спектр Фурье, показанный на предыдущем рисунке справа, построен на отрезке одного периода. Из формулы (13) легко видеть, что функция  $F(u)$  имеет период  $M$ , т.е.  $F(u + M) = F(u)$ . Следовательно, этот период имеет и спектр. Поэтому его расширенный на более широкий интервал график имеет максимум в окрестности начала координат. Часто желательно наблюдать этот график на отрезке, симметричном относительно начала координат, так как показано на следующем рисунке справа



Для сдвига спектра в середину отрезка [1 256] надо исходную функцию умножить на  $(-1)^x$ . Напомним, что здесь  $x$  целое число. Следующие команды выполняют это действие.

```
% продолжение сценария ex28.m
fx1=fx.*(-1).^x;
fu1=(fx1*exp(-i*2*pi*u'*x/M))/M;
figure, plot(u,abs(fu1),'LineWidth',2);
axis([1 M 0 max(abs(fu1))]);
grid on;
```



Здесь мы использовали следующее свойство ДПФ: чтобы получить спектр исходной функции  $f(x)$  сдвинутый в середину диапазона  $[1, M]$  надо построить спектр функции  $g(x) = (-1)^x f(x)$ . Проверим это утверждение. Для этого нам потребуется свойство сдвига ДПФ в виде

$$f(x)\exp(i2\pi u_0 x/M) \Leftrightarrow F(u - u_0)$$

Действительно, пусть  $g(x) = f(x)\exp(i2\pi u_0 x/M)$ . Тогда ее ДПФ будет

$$\begin{aligned} G(u) &= \frac{1}{M} \sum_{x=0}^{M-1} g(x) e^{-i\frac{2\pi u x}{M}} = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{i\frac{2\pi u_0 x}{M}} e^{-i\frac{2\pi u x}{M}} = \\ &= \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-i\frac{2\pi(u-u_0)x}{M}} = F(u - u_0) \end{aligned}$$

При  $u_0 = M/2$  мы получаем  $\exp(i2\pi u_0 x/M) = \exp(i\pi x) = (-1)^x$ . Тогда

$$f(x)(-1)^x \Leftrightarrow F(u - M/2)$$

Т.о., чтобы увидеть спектр функции  $f(x)$ , сдвинутый вправо на середину отрезка  $[0, M-1]$ , надо построить спектр функции  $f(x)(-1)^x$ .

Все рассмотренные ранее формулы относятся к случаю, когда переменная  $u$  изменяется в диапазоне  $[0, M-1]$ . При реализации в MatLab эта переменная будет принимать значения от  $u=1$  до  $M$ . При этом реальный центр спектра Фурье будет находиться в точке  $M/2+1$ . Для того, чтобы обеспечить целочисленность координат после сдвига, будем требовать чтобы  $M$  было четным.

## 4.2 Двумерное дискретное преобразование Фурье

Аналогично одномерному ДПФ, можно получить формулы двумерного ДПФ. Прямое двумерное дискретное преобразование Фурье преобразует дискретную функцию 2 – х переменных (изображение)  $f(x,y)$ , заданную в пространственной координатной системе  $(x,y)$ , в двумерную дискретную функцию, заданную в частотной координатной системе  $(u,v)$

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

Поскольку  $f_{cp} = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) = \frac{1}{M \cdot N} F(0,0)$ , то величина  $F(0,0)$  равна числу  $M \cdot N$ , умноженному на среднее значение функции  $f(x,y)$ .

Обратное дискретное преобразование Фурье (ОДПФ) имеет вид:

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

Диапазоны изменения пространственных координат  $x=0,1,2,\dots,M-1$ , и  $y=0,1,2,\dots,N-1$  и частотных  $u=0,1,2,\dots,M-1$ , и  $v=0,1,2,\dots,N-1$  одинаковые. Прямоугольную область размера  $M \times N$  принято называть частотным прямоугольником. Т.о. частотный прямоугольник имеет те же размеры, что и исходное изображение. Как и в одномерном случае, положение множителя  $1/M \cdot N$  не имеет значения. Он может быть отнесен в обратное преобразование или разделен на два сомножителя  $1/\sqrt{MN}$  в прямом и обратном преобразованиях.

Таким образом, зная  $F(u,v)$ , можно восстановить  $f(x,y)$  с помощью обратного ДПФ. Величины  $F(u,v)$  в этих уравнениях принято называть коэффициентами разложения Фурье, количество которых равно  $M \times N$ .

Даже если изображение  $f(x,y)$  вещественно, его преобразование Фурье является, как правило, комплексным. Основной метод визуального анализа этого преобразования заключается в вычислении его спектра (т.е. абсолютной величины  $F(u,v)$ ) и его отображения на дисплее. Если  $R(u,v)$  и  $I(u,v)$  обозначают вещественную и мнимую компоненты  $F(u,v)$

$$F(u,v) = R(u,v) + i \cdot I(u,v)$$

то спектр Фурье задается выражением

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{1/2}.$$

Величины

$$\phi(u,v) = \text{arctg} \left[ \frac{I(u,v)}{R(u,v)} \right]$$

называются фазовым спектром изображения.

Энергетический спектр (или спектральная функция) равен квадрату модуля спектра

$$P(u, v) = |F(u, v)|^2 = R^2(u, v) + I^2(u, v).$$

С точки зрения визуализации не принципиально, какую функцию отображать на экране,  $F(u, v)$  или  $P(u, v)$ .

Приведем некоторые свойства ДПФ. Пара преобразований Фурье обладают следующими трансляционными свойствами (свойствами по отношению к сдвигам)

$$f(x, y)e^{i2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0) \quad (3)$$

и

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-i2\pi(u_0x/M + v_0y/N)} \quad (4)$$

где двойная стрелка использована для обозначения того, что соответствующие функции образуют фурье пару. Действительно, пусть

$g(x, y) = f(x, y)\exp\left(i2\pi\left(\frac{u_0x}{M} + \frac{v_0y}{N}\right)\right)$ . Тогда ее ДПФ будет

$$\begin{aligned} G(u, v) &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x, y) e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} = \\ &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{i2\pi\left(\frac{u_0x}{M} + \frac{v_0y}{N}\right)} e^{-i2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} = \\ &= \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi\left(\frac{(u-u_0)x}{M} + \frac{(v-v_0)y}{N}\right)} = F(u - u_0, v - v_0) \end{aligned}$$

При  $u_0 = M/2$ ,  $v_0 = N/2$  мы получаем

$$\exp\left(i2\pi\left(\frac{u_0x}{M} + \frac{v_0y}{N}\right)\right) = \exp(i\pi(x + y)) = (-1)^{x+y}$$

Тогда

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2) \quad (5)$$

Аналогично проверяется соотношение (4).

Функция ДПФ является периодической по обоим переменным  $u$  и  $v$ , периоды которой равны  $M$  и  $N$  соответственно.

$$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N). \quad (6)$$

Действительно, каждое слагаемое суммы (1) содержит множитель

$c(u, v) = \exp\left(-2\pi i\left(\frac{ux}{M} + \frac{vy}{N}\right)\right)$ , который является периодической функцией по

каждой переменной  $u$  и  $v$ . Например,

$$c(u+M, v) = \exp\left(-2\pi i \left(\frac{(u+M)x}{M} + \frac{vy}{N}\right)\right) = \exp\left(-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)\right) \cdot \exp(-2\pi ix) = \\ = \exp\left(-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)\right) = c(u, v),$$

поскольку для целых  $x$  имеем  $\exp(-2\pi ix) = 1$ . Аналогично проверяются другие равенства (6).

Если  $f(x, y)$  вещественная, то ее преобразование Фурье является сопряжено симметричным относительно начала координат, т. е.

$$F(u, v) = \overline{F(-u, -v)} \quad (7)$$

Это следует из того, что каждый множитель  $c(u, v)$  обладает этим свойством

$$c(u, v) = \exp\left(-2\pi i \left(\frac{ux}{M} + \frac{vy}{N}\right)\right) = \overline{\exp\left(-2\pi i \left(\frac{-ux}{M} + \frac{-vy}{N}\right)\right)} = \overline{c(-u, -v)}$$

Из (7) следует симметрия спектра Фурье относительно начала координат:

$$|F(u, v)| = |F(-u, -v)|.$$

Обратное преобразование Фурье также имеет свойство периодичности:

$$f(x, y) = f(x+M, y) = f(x, y+N) = f(x+M, y+N).$$

Обычной практикой перед визуализацией спектра стало умножение исходной функции (изображения) на  $(-1)^{x+y}$ , т.е. построение спектра не исходной функции, а функции  $f(x, y)(-1)^{x+y}$ . Как мы видели (см. формулу (5)), это просто сдвигает спектр в середину спектрального прямоугольника. Т.е. та точка, где значение преобразования Фурье равно  $F(0,0)$  находится в точке с координатами  $u=M/2$ ,  $v=N/2$ . При вычислениях преобразования Фурье в Matlab, индексы принимают значения от  $u=1$  до  $u=M$  и от  $v=1$  до  $v=N$ . В этом случае центр фурье – образа находится в точке  $u = M/2 + 1$  и  $v = N/2 + 1$ . Для того, чтобы обеспечить целочисленность координат после сдвига, будем требовать, чтобы числа  $M$  и  $N$  были четными. В конкретных реализациях вычисляется только один период ДПФ. Поэтому мы будем работать с массивами размера  $M \times N$ .

### 4.3 Вычисление и визуализация дискретного преобразования Фурье

Прямое и обратное ДПФ на практике вычисляются с помощью алгоритма быстрого преобразования Фурье (FFT, Fast Fourier Transform). Для прямоугольного дискретного изображения  $f$  (матрицы, массива) размером  $M \times N$  алгоритм FFT реализован в Matlab функцией `fft2`, которая имеет следующий синтаксис:

$$F = \text{fft2}(f).$$

Эта функция возвращает преобразование Фурье размера  $M \times N$ . Для получения спектра Фурье следует применить команду

$$S = \text{abs}(F),$$

которая вычисляет абсолютную величину каждого элемента массива F.

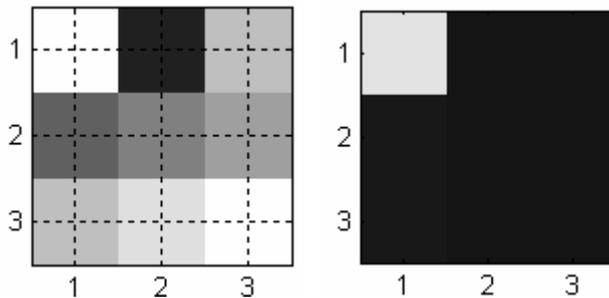
### Пример 1.

```
A=[9 2 7; 4 5 6; 7 8 9]
```

```
A =
```

```
     9     2     7
     4     5     6
     7     8     9
```

```
image(A); colormap(gray(9)); % изображение слева
set(gcf,'Color',[1 1 1]); axis image;
grid on
```



Напомним, что двумерный массив функцией `image(...)` интерпретируется как индексированное изображение, а трехмерный с элементами в диапазоне от 0 до 1 – как RGB изображение.

```
F = fft2(A)
```

```
F =
```

```
 57.0000          1.5000 + 6.0622i    1.5000 - 6.0622i
-1.5000 + 7.7942i    6.0000 + 3.4641i    6.0000 - 3.4641i
-1.5000 - 7.7942i    6.0000 + 3.4641i    6.0000 - 3.4641i
```

```
S = abs(F)
```

```
S =
```

```
 57.0000    6.2450    6.2450
  7.9373    6.9282    6.9282
  7.9373    6.9282    6.9282
```

```
image(S); colormap(gray) % рисунок вверху справа
axis image; set(gcf,'Color',[1 1 1]);
```

### Пример 2. ex17.m

```
close all;
```

```
B=zeros(21);
```

```
BC=ones(5);
```

```
B(9:13,9:13)=BC;
```

```
B3=cat(3,B,B,B); % создаем 3-х мерный массив, чтобы
                  % функция image интерпретировала его как
                  % RGB изображение
```

```
image(B3); % левое изображение
```

```
axis image; set(gcf,'Color',[1 1 1]);
```

```
% выполняем ДПФ и отображаем его
```

```
F=fft2(B); S=abs(F); n=max(max(S));
```

```
S1=S/n; % создаем 3-х мерный массив с
```

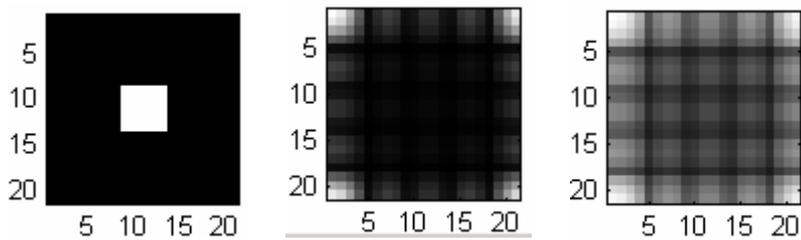
```
S3=cat(3,S1,S1,S1); % диапазоном элементов от 0 до 1
```

```
figure,image(S3); % среднее изображение
```

```
axis image; set(gcf,'Color',[1 1 1]);
```

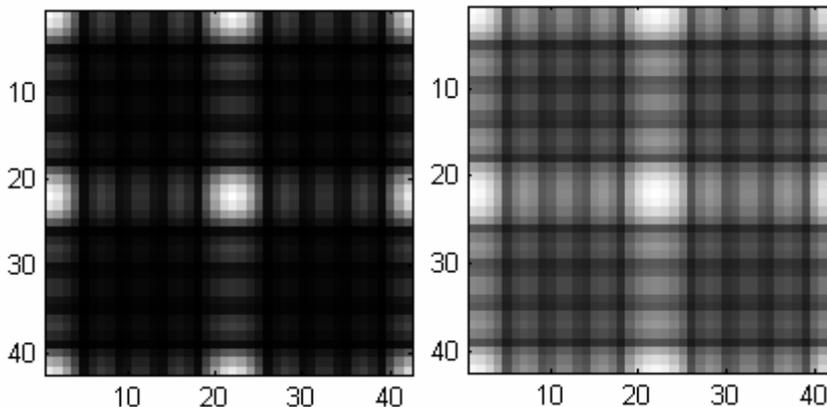
```
% выполняем преобразование яркости для ПФ
```

```
SL3=imadjust(S3,[],[],0.38);
figure,image(SL3); % рисунок справа
axis image; set(gcf,'Color',[1 1 1]);
```



Часто преобразование Фурье выглядит так, как на среднем или правом рисунке – с яркими областями по углам изображения. Поскольку изображение периодически по обеим координатам, то на самом деле эти яркие области прилегают друг к другу. Рассмотрим это изображение, но содержащее два периода по каждой координате.

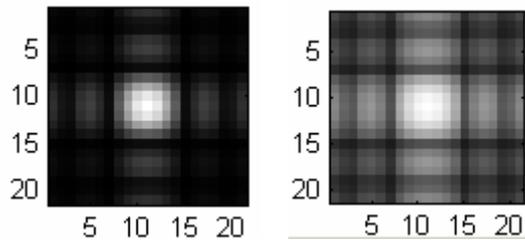
```
% продолжение сценария ex17.m
% расширяем изображение в 2 раза по обеим направлениям
C=zeros(42);
C(1:21,1:21)=S1; C(1:21,22:42)=S1;
C(22:42,1:21)=S1; C(22:42,22:42)=S1;
C3=cat(3,C,C,C); figure,image(C3);
axis image; set(gcf,'Color',[1 1 1]);
% строим преобразованное изображение
CL3=imadjust(C3,[],[],0.38);
figure,image(CL3);
axis image; set(gcf,'Color',[1 1 1]);
```



Обычно нагляднее наблюдать центрированное изображение ДПФ, которое можно получить из предыдущих рисунков выделением средней части расширенного изображения.

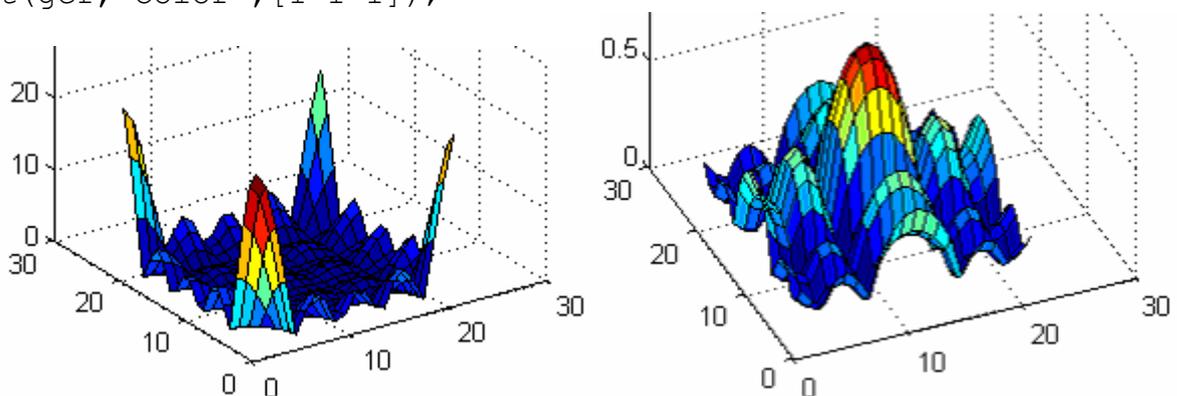
```
% продолжение сценария ex17.m
% выделение центральной части увеличенного изображения
CC=C(12:32,12:32);
CC3=cat(3,CC,CC,CC);
figure,image(CC3); % центрированное слева
axis image; set(gcf,'Color',[1 1 1]);

CCL3=CL3(12:32,12:32,:);
figure,image(CCL3); % центрированное преобразованное справа
axis image; set(gcf,'Color',[1 1 1]);
```



Интересно посмотреть на спектр, как на поверхность

```
% продолжение сценария ex17.m
% отобразить спектр как поверхность
figure, surface(S);grid on;
set(gcf,'Color',[1 1 1]);
% отобразить центрированный логарифмированный спектр
% как поверхность
figure, surface(CCL3(:,:,1));grid on;
set(gcf,'Color',[1 1 1]);
```



Чтобы не выполнять вручную смещение в пакете ИРТ есть функция `fftshift`, выполняющая смещение начала координат в центр частотной области. Ее синтаксис имеет вид

$F_c = \text{fftshift}(F)$ ,

где  $F$  — это преобразование, вычисленное с помощью `fft2`, а  $F_c$  — отцентрированное преобразование. Функция `fftshift` переставляет квадранты  $F$  надлежащим способом. Например, пусть

```
A=[1 2 3 4; 6 7 8 9 ; 11 12 13 14 ; 16 17 18 19]
```

A =

```
1     2     3     4
6     7     8     9
11    12    13    14
16    17    18    19
```

Создадим увеличенную в 2 раза повторяющуюся матрицу B

```
B=zeros(8); B(1:4,1:4)=A; B(1:4,5:8)=A;
```

```
B(5:8,1:4)=A; B(5:8,5:8)=A
```

B =

```
1     2     3     4     1     2     3     4
6     7     8     9     6     7     8     9
11    12    13    14    11    12    13    14
16    17    18    19    16    17    18    19
1     2     3     4     1     2     3     4
6     7     8     9     6     7     8     9
11    12    13    14    11    12    13    14
16    17    18    19    16    17    18    19
```

Выделим из В центральную часть матрицы

```
C=zeros(4); C=B(3:6,3:6)
```

```
C =  
    13    14    11    12  
    18    19    16    17  
     3     4     1     2  
     8     9     6     7
```

Сравним результат с результатом работы функции `fftshift`

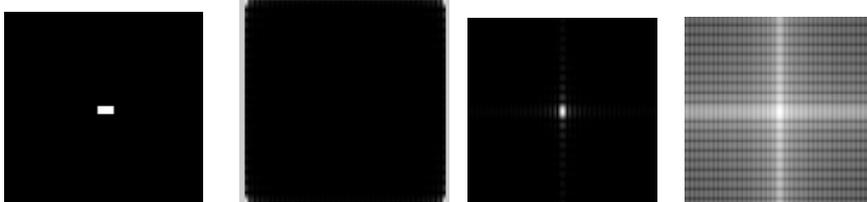
```
fftshift(A)
```

```
ans =  
    13    14    11    12  
    18    19    16    17  
     3     4     1     2  
     8     9     6     7
```

Визуальный анализ спектра при отображении его в виде изображения является важным инструментом при работе в частотной области.

**Пример 3.** Рассмотрим простое изображение `f` и построим его спектр с помощью следующей последовательности команд (`ex18.m`):

```
f=imread('SimpleImage.tif');  
imshow(f); % изображение слева  
F = fft2(double(f)); S = abs(F);  
imshow(S, [ ]); % рисунок второй слева; яркость в уголках  
Fc = fftshift(F); % сдвиг  
imshow(abs(Fc), [ ]); % изображение третье слева  
S2 = log(1 + abs(Fc)); % логарифмическое преобразование  
imshow(S2, [ ]); % изображение справа
```



На втором слева рисунке приведен результат ДПФ. Четыре яркие точки по углам изображения появляются в силу свойства периодичности. В этом примере с помощью команд

```
Fc = fftshift(F); imshow(abs(Fc), [ ]);
```

мы получили изображение с ярким пятном в центре. Хотя сдвиг спектра был сделан корректно, динамический диапазон его значений весьма широк (от 0 до 204000). Поэтому изображение довольно темное.

```
max(max(Fc))
```

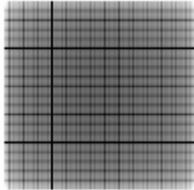
```
ans =  
    204000
```

Эту проблему мы решили с помощью логарифмического преобразования, выполнив команды

```
S2 = log(1 + abs(Fc)); imshow(S2, [ ]);
```

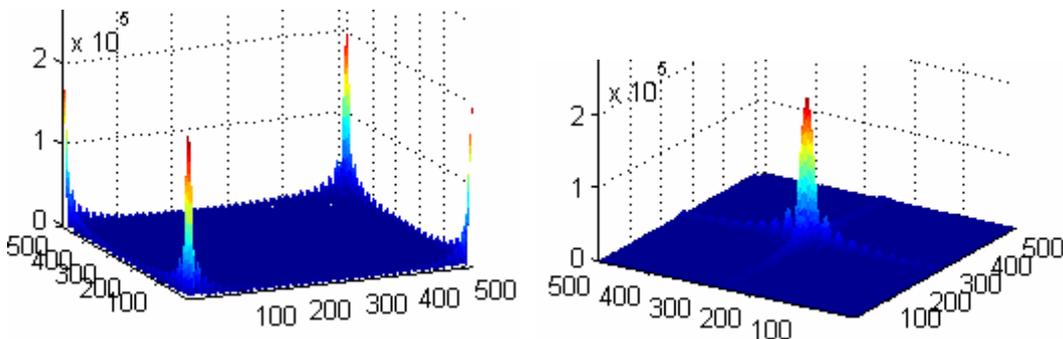
Функция `ifftshift` обращает центрирование спектра. Ее синтаксис имеет вид `f2=ifftshift(Fc)`. В нашем примере мы ее можем использовать следующим образом

```
s2=ifftshift(S2); imshow(s2,[]);
```



Теперь посмотрим на поверхность спектра и центрированного спектра

```
% продолжение сценария ex18.m
% отобразить спектр как поверхность
figure, surface(S, 'EdgeColor', 'none'); grid on;
set(gcf, 'Color', [1 1 1]); axis([1 512 1 512]);
% отобразить центрированный спектр как поверхность
figure, surface(abs(Fc), 'EdgeColor', 'none'); grid on;
set(gcf, 'Color', [1 1 1]); axis([1 512 1 512]);
```



Обратное преобразование Фурье вычисляется с помощью функции `ifft2`, синтаксис которой имеет вид

```
f = ifft2(F),
```

где  $F$  — это преобразование Фурье, а  $f$  — соответствующее изображение. Если исходное изображение, использованное при вычислении  $F$ , было вещественным, то и обратное преобразование, в теории, должно быть вещественным. Однако на практике выход `ifft2` часто имеет очень малую мнимую компоненту, которая получается из-за ошибок округления при выполнении арифметических действий с плавающей запятой. Поэтому хорошим практическим советом является выделение вещественной части после выполнения обратного преобразования Фурье. Эти два действия можно выполнить в одной команде:

```
f = real(ifft2(F));
```

Следующая последовательность команд строит исходное изображение, ДПФ и строит изображение, полученное ОДПФ.

```
f=imread('SimpleImage.tif'); imshow(f);
F = fft2(double(f)); % преобразование типа - double(uint8)
imshow(abs(F),[]);
f2 = ifft2(F); imshow(f2); % тот же прямоугольник
```

Обратите внимание на тип изображения  $F$  – он комплексный (`double (complex)`).

Рассмотрим матрицу  $A$ , взятую нами в примере 1.

```
close all;
A=[9 2 7; 4 5 6;7 8 9]
image(A); colormap(gray(9));
F=fft2(A)
S=abs(F)
S1=S/max(max(S));
S3=cat(3,S1,S1,S1);
figure,image(S3);
f = real(iff2(F))
figure,image(f); colormap(gray(9)); axis image
```

В результате обращения ДПФ мы получаем матрицу  $f$ , которая полностью совпадает с исходной матрицей  $A$ .

Есть еще один формат вызова функции ДПФ

```
Y = fft2(X,m,n);
```

который усекает или дополняет  $X$  нулями для создания массива  $m \times n$  перед преобразованием. Результат также имеет размер  $m \times n$ .

```
A=[1 2 3; 4 5 6; 7 8 9]
```

```
A =
```

```
     1     2     3
     4     5     6
     7     8     9
```

```
F1=fft2(A)
```

```
F1 =
```

```
 45.0000          -4.5000 + 2.5981i  -4.5000 - 2.5981i
-13.5000 + 7.7942i           0           0
-13.5000 - 7.7942i           0           0
```

```
F2=fft2(A,4,4)
```

```
F2 =
```

```
 45.0000          -6.0000 -15.0000i  15.0000          | -6.0000 +15.0000i
-18.0000 -15.0000i  -5.0000 + 8.0000i  -6.0000 - 5.0000i  5.0000 - 4.0000i
 15.0000          -2.0000 - 5.0000i   5.0000          | -2.0000 + 5.0000i
-18.0000 +15.0000i  5.0000 + 4.0000i  -6.0000 + 5.0000i -5.0000 - 8.0000i
```

Вот как изображение расширилось

```
f2 = ifft2(F2)
```

```
f2 =
```

```
     1     2     3     0
     4     5     6     0
     7     8     9     0
     0     0     0     0
```

Как видим, исходная матрица снизу и справа дополнилась нулями. Вот как изображение усекается

```
F3=fft2(A,2,2)
```

```
F3 =
```

```
 12    -2
  -6     0
```

```
f3 = ifft2(F3)
```

```
f3 =
```

```
     1     2
     4     5
```

Матрица всегда расширяется справа и снизу.

Для тренировки выполним ДПФ одномерного вектора

```
A=[1 2 3]
```

```
F=fft2(A)
```

```
f = real(iff2(F))
```

Обратное ДПФ  $f$  оказалось точно равным исходному вектору  $A$ .

```
A = 1 2 3
```

```
F = 6.0000 -1.5000 + 0.8660i -1.5000 - 0.8660i
```

```
f = 1 2 3
```

**Пример.** В качестве тренировки использования функций Matlab для одномерных функций построим спектр одномерной функции, рассмотренной нами ранее в примере 2 п 4.1.

```
% сценарий ex31.m
```

```
close all;
```

```
M=256;
```

```
x=1:M; u=1:M;
```

```
fx=zeros(1,M);
```

```
fx(1:M/32)=1;
```

```
subplot(2,2,1); plot(x,fx,'LineWidth',2); % исходная функция
```

```
axis([1 M 0 max(abs(fx*1.1))]); grid on;
```

```
set(gcf,'Color',[1 1 1]);
```

```
fu=(fx*exp(-i*2*pi*u'*x/M))/M;
```

```
subplot(2,2,2);plot(u,abs(fu),'LineWidth',2); % ПФ
```

```
axis([1 M 0 max(abs(fu))]); grid on;
```

```
F1=fft2(fx);
```

```
S=abs(F1);
```

```
subplot(2,2,3);plot(u,S,'LineWidth',2); % нецентрированный спектр
```

```
axis([1 M 0 max(S)]); grid on;
```

```
Fc = fftshift(S);
```

```
subplot(2,2,4);plot(u,Fc,'LineWidth',2); % центрированный спектр
```

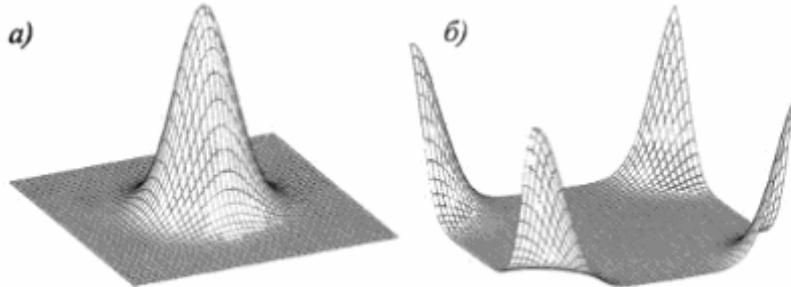
```
axis([1 M 0 max(S)]); grid on;
```

Спектр и центрированный спектр, полученные с использованием функций  $\text{fft2}(fx)$  и  $\text{fftshift}(S)$ , естественно оказались идентичными примеру 2.

#### 4.4 Фильтрация в частотной области.

После получения ДПФ  $F(u,v)$  изображения  $f(x,y)$  его (ДПФ) часто подвергают изменению. Например, требуется уменьшить коэффициенты (элементы матрицы  $F(u,v)$ ) отвечающие высоким частотам. Для этого функцию / матрицу  $F(u,v)$  умножают на другую функцию / матрицу подходящего размера, которая называется передаточной функцией. Такое действие называют фильтрацией, а саму передаточную функцию обычно называют фильтром. Основная идея фильтрации в частотной области заключается в подборе передаточной функции, которая модифицирует

$F(u, v)$  специфическим образом. Например, передаточная функция на следующем рисунке (а) будучи умноженной на центрированную функцию  $F(u, v)$ , ослабляет компоненты  $F(u, v)$ , лежащие далеко от центра и оставляет срединные значения мало изменившимися. Фильтр с такими характеристиками называется низкочастотным. На правом рисунке (б) приведен тот же фильтр после его сдвига функцией `fftshift`. Такой формат фильтров особенно часто используется, когда преобразования Фурье не центруются.



Передаточные функции а) центрированного низкочастотного фильтра; б) в формате, используемом при фильтрации

Идеальный низкочастотный фильтр имеет передаточную функцию

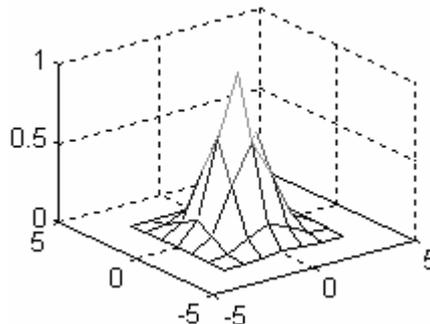
$$H(u, v) = \begin{cases} 1 & \text{при } D(u, v) \leq D_0 \\ 0 & \text{при } D(u, v) > D_0, \end{cases} \quad (1)$$

где  $D_0$  — это заданное 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. Помня о том, что фильтр  $H$  умножается на преобразование Фурье изображения, можно заключить, что идеальный фильтр «срезает» (умножает на ноль) все компоненты  $F$ , лежащие вне этого круга, и оставляет неизменными (умножает на 1) все компоненты, находящиеся внутри или на границе окружности.

График поверхности фильтра строится так же как график поверхности любой функции 2-х переменных. Например, для функции  $\exp\left(-\frac{u^2 + v^2}{2D_0^2}\right)$

график на грубой сетке можно построить следующими командами

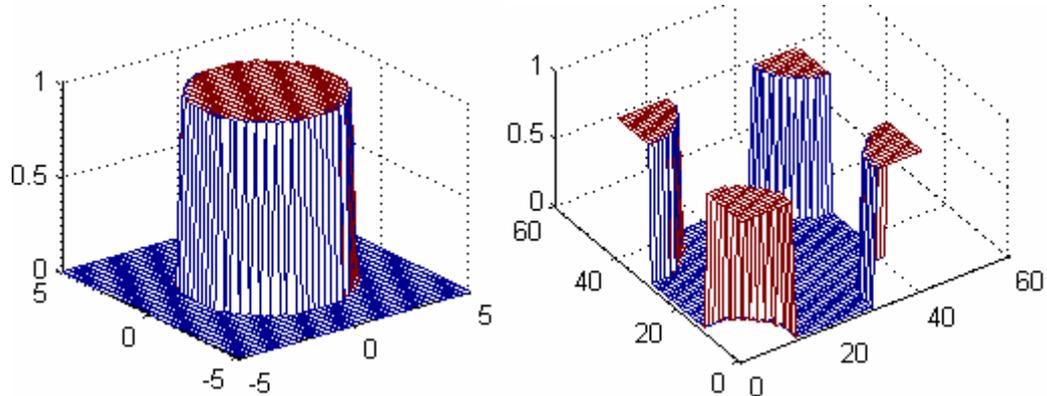
```
H=@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2)));
[U,V]=meshgrid(-3:1:3)
HH=H(U,V,1)
mesh(U,V,HH);
```



Для работы с изображениями нам потребуются нецентрированные фильтры. Для их построения можно использовать функцию `ifftshift(...)`.

Вот какой вид имеет идеальный низкочастотный фильтр при  $D_0=3$ .

```
HE=@(U,V,D0) (1.*(U.^2 + V.^2)<D0^2));
[U,V]=meshgrid(-5:0.25:5);HHE=HE(U,V,3);mesh(U,V,HHE);
% построение нецентрированного фильтра
HHS=ifftshift(HHE); figure,mesh(HHS);
```



Обратите внимание, что нецентрированный фильтр построен на целочисленной координатной сетке, такой как нам потребуется в дальнейшем.

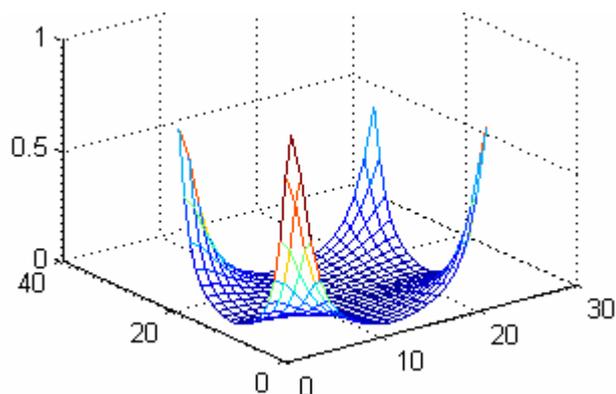
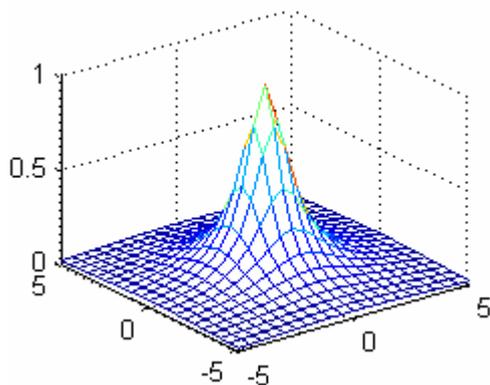
Низкочастотный фильтр Баттерворта (BLPF, Butterworth LowPass Filter) порядка  $n$  с обрезанием частот на расстоянии  $D_0$  от начала координат имеет передаточную функцию

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

В отличие от идеального низкочастотного фильтра, функция фильтра BLPF не имеет разрыва в пороговой точке  $D_0$ . Для фильтров с гладкой передаточной функцией принято задавать частоту среза, которая определяется положением точек, для которых функция  $H(u,v)$  меньше определенной доли ее максимального значения. В предыдущем уравнении, когда  $D(u,v) = D_0$  значение передаточной функции  $H(u,v) = 0.5$  (т.е., 50% от максимального значения, которое равно 1)

Вот какой вид имеет низкочастотный фильтр Баттерворта при  $D_0=1$  и  $n=1$ .

```
HB=@(U,V,D0,n) 1./(1+((U.^2 + V.^2)./D0^2).^n);
[U,V]=meshgrid(-5:0.5:5);HHB=HB(U,V,1,1);mesh(U,V,HHB);
% построение нецентрированного фильтра
HHS=ifftshift(HHB); figure,mesh(HHS);
```



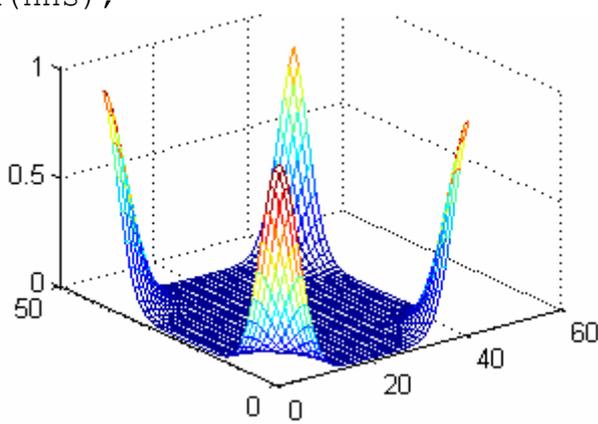
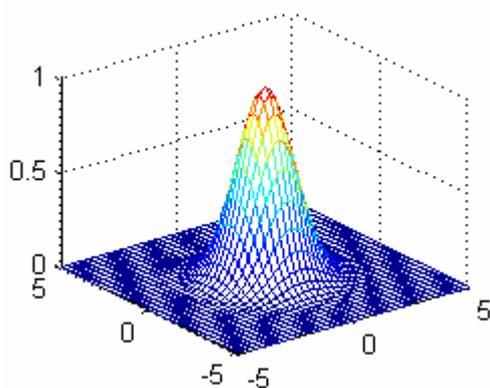
Передаточная функция гауссова низкочастотного фильтра (GLPF, Gaussian LowPass Filter) задается формулой

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

В точках, для которых  $D(u, v) = D_0$ , значение фильтра составляет 0.6065 от максимального значения, которое равно 1.

Вот какой вид имеет гауссов низкочастотный фильтр при  $D_0=1$

```
H=@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2)));
[U,V]=meshgrid(-5:0.25:5);HH=H(U,V,1);mesh(U,V,HH);
% построение нецентрированного фильтра
HHs=ifftshift(HH); figure,mesh(HHs);
```



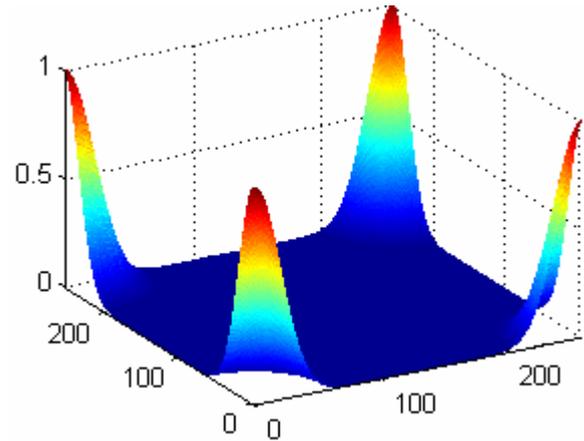
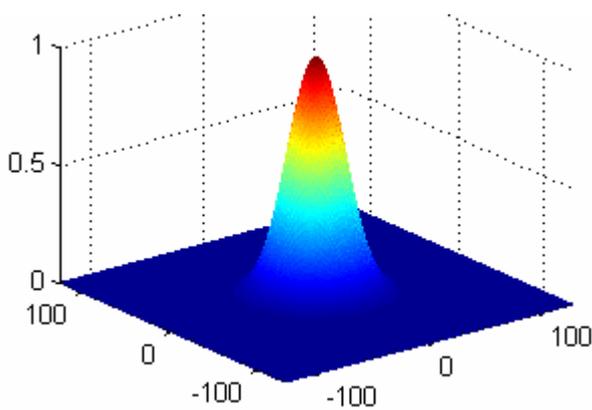
Если размер изображения равен  $M \times N$ , то для построения его нецентрированного фильтра нужно будет построить соответствующий фильтр на прямоугольнике  $[-M/2 .. M/2, -N/2..N/2]$  (полагаем, что  $M$  и  $N$  четные). Например, если мы собираемся обрабатывать изображение SimpleImage2.tif с использованием Гауссова фильтра, то надо прочитать изображение, определить его размеры, которые совпадут с размерами матрицы его преобразования Фурье, а потом строить фильтр

```
f=imread('SimpleImage2.tif');imshow(f);
[M,N]=size(f)
M =
    256
N =
    256
H=@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2)));
[U,V]=meshgrid(-M/2:M/2,-N/2:N/2);
HH=H(U,V,20);mesh(U,V,HH);
set(gcf,'Color',[1 1 1]); axis([-M/2 M/2 -N/2 N/2]);
```

```

HHs=ifftshift(HH); figure, mesh(HHs);
axis([0 M 0 N]);

```



Таким образом, матрица  $HHs$  в этом примере будет представлять нецентрированный фильтр Гаусса.

Предложенная процедура сдвига потребует некоторого уточнения, если числа  $M$  и  $N$  будут нечетными. Для центрирования нам понадобится следующий алгоритм. При нечетном  $M$  он создает сдвинутый влево следующий центрированный массив

```

M=9; s=floor(M/2)-floor((M-1)/2), u=1:M,
v=-floor((M-1)/2)-s: floor((M-1)/2)
s =

```

```

    0
u =
    1     2     3     4     5     6     7     8     9
v =
   -4    -3    -2    -1     0     1     2     3     4

```

При четном  $M$  имеем

```

M=8; s=floor(M/2)-floor((M-1)/2), u=1:M,
v=-floor((M-1)/2)-s: floor((M-1)/2) % строим вектор
s =
    1
u =
    1     2     3     4     5     6     7     8
v =
   -4    -3    -2    -1     0     1     2     3

```

Частотная фильтрация состоит в преобразовании матрицы  $F(u,v)$  ДПФ, например, в умножении на матрицу низкочастотного фильтра, и последующем применении ОДПФ, т.е. получении прообраза Фурье преобразованного спектра.

**Пример.** Используя описанную выше сдвижку, посмотрим, что делает частотная фильтрация с очень простым изображением, составленным из белой и черной полос одинаковой ширины

```

% сценарий ex25.m
close all;
f=zeros(8,8); f(1:4,:)=1;
image(cat(3,f,f,f)); % изображение после сценария слева
set(gcf,'Color',[1 1 1]);axis image;

```

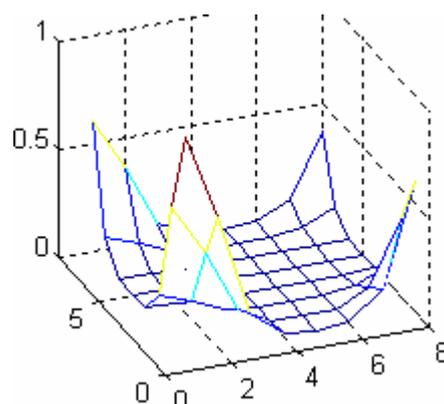
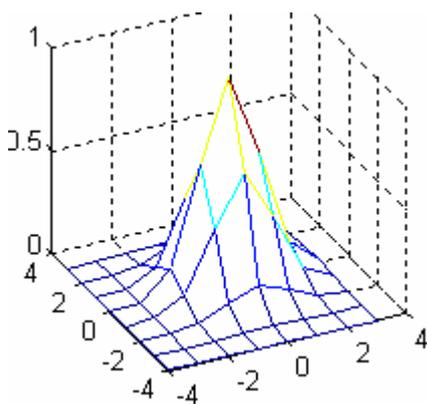
```

F = fft2(double(f)); % Преобразование Фурье
[M, N] = size(f);

% функция Гаусса (центрированный гауссов фильтр)
H=@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2)));
% создание матрицы центрированного Гауссова
% низкочастотного фильтра размера M x N
sM=floor(M/2)-floor((M-1)/2);
MM=-floor((M-1)/2)-sM: floor((M-1)/2);
sN=floor(N/2)-floor((N-1)/2);
NN=-floor((N-1)/2)-sN: floor((N-1)/2);
[U,V]=meshgrid(MM,NN);
HH=H(U,V,1); % пороговое значение 1
figure, mesh(U,V,HH); % образ слева
set(gcf,'Color',[1 1 1]); axis([-M/2 M/2 -N/2 N/2]);

% децентрирование Гауссова фильтра
HHs=ifftshift(HH);
figure, mesh(HHs); % образ справа
set(gcf,'Color',[1 1 1]); axis([0 M 0 N]);

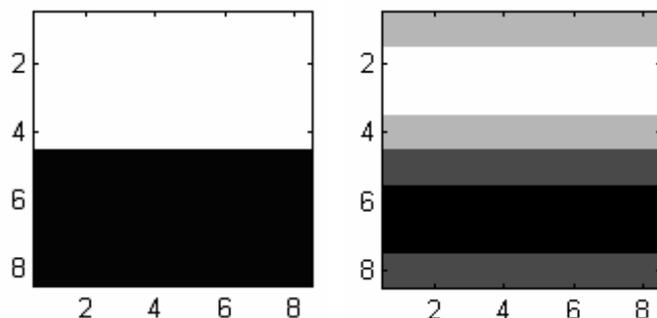
```



```

G = HHs.*F; % фильтрация в частотной области
g = real(ifft2(G)); % обращение отфильтрованного спектра
% создание 3-х мерного массива со значениями в диапазоне
% от 0 до 1
g1=(g-min(min(g)))/(max(max(g))-min(min(g)));
figure,image(cat(3,g1,g1,g1)); % изображение после фильтрации
% плохо видна серая горизонтальная полоса внизу
set(gcf,'Color',[1 1 1]); axis image;

```



Слева исходное изображение, справа отфильтрованное. Плохо видна серая горизонтальная полоса внизу

Поскольку размер изображения был всего 8 x 8 точек, то результат не очень похож на оригинал, однако сохранил основные его свойства – наличие двух горизонтальных полос, белой и черной.

**Пример.** Вот пример фильтрации подобного изображения, но размером 256 x 256 точек. Для него мы составили следующий сценарий (ex26.m)

```
close all;
f=imread('SimpleImage2.tif');
imshow(f, []);
axis image;

F = fft2(double(f));
[M, N] = size(f);

% создание центрированного Гауссова фильтра размера M x N
H=@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2))); % гауссов фильтр

sM=floor(M/2)-floor((M-1)/2);
MM=-floor((M-1)/2)-sM: floor((M-1)/2);
sN=floor(N/2)-floor((N-1)/2);
NN=-floor((N-1)/2)-sN: floor((N-1)/2);
[U,V]=meshgrid(MM,NN);
HH=H(U,V,20); % % пороговое значение 20
figure, mesh(U,V,HH);
set(gcf,'Color',[1 1 1]); axis([-M/2 M/2 -N/2 N/2]);

% децентрирование Гауссова фильтра
HHs=ifftshift(HH);
figure, mesh(HHs);
set(gcf,'Color',[1 1 1]); axis([0 M 0 N]);

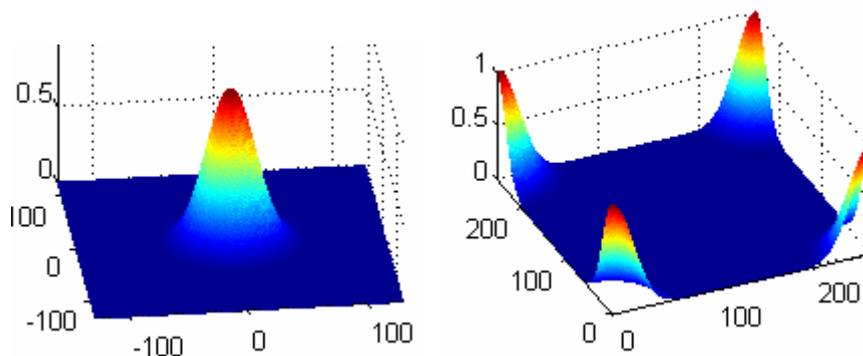
G = HHs.*F; % фильтрация в частотной области
g = real(ifft2(G)); % обращение отфильтрованного спектра

figure, imshow(g, [ ]); % плохо видна серая горизонтальная
% полоса внизу
set(gcf,'Color',[1 1 1]); axis image;
```

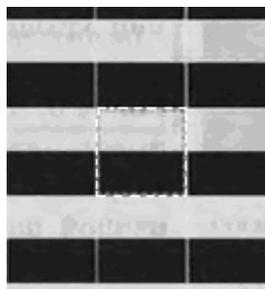


Исходное изображение слева, восстановленное при  $D_0=5$  в середине, восстановленное при  $D_0=1$  справа. С увеличением  $D_0$  фильтр становится уже, отфильтровывается больше частот, ширина размытых границ увеличивается. Наоборот, увеличивая  $D_0$ , мы оставляем больше высоких частот, и изображение менее размыто.

На следующем рисунке показаны центрированный (слева) и децентрированный фильтр Гаусса при  $D_0=20$ .



Размытие только горизонтальных границ можно пояснить следующим рисунком, на котором графически изображено воображаемое периодическое повторение изображения. Исходное изображение показано пунктиром. В тех случаях, когда черная и белая области смешиваются в фильтре, результатом будет серый (размытый) вид. Правые и левые границы воображаемого изображения продолжают идентичными цветами, поэтому смазывания левой и правой границ изображения нет.



Таким образом, для фильтрации изображения в частотной области мы умножали ДПФ изображения на соответствующую матрицу – фильтр, а затем вычисляли обратное ДПФ. Если бы матрица – фильтр была единичной (не меняла бы ДПФ), обратное ДПФ восстановило бы исходное изображение.

Основой линейной фильтрации в частотной и пространственной области служит теорема о свертке, которую можно сформулировать так:

$$f(x, y) * h(x, y) \iff H(u, v)F(u, v) \quad (4)$$

и в обратную сторону

$$f(x, y)h(x, y) \iff H(u, v) * F(u, v). \quad (5)$$

Здесь символ «\*» обозначает операцию свертки двух функций, а выражения по обе стороны от двойных стрелок определяют соответствующие пары при преобразовании Фурье. Например, первое выражение означает, что свертку двух пространственных функций можно получить, если вычислить обратное преобразование Фурье от произведения прямых преобразований Фурье этих двух функций.

*Замечание.* Здесь можно было бы доказать формулы (4) и (5) для ДПФ.

Фильтрация в пространственной области состоит из свертки изображения  $f(x, y)$  и маски  $h(x, y)$ . Линейная пространственная свертка объяснялась ранее. В соответствии с теоремой о свертке, тот же результат можно получить обратным ДПФ от произведения  $F(u, v)$  на  $H(u, v)$  —

преобразование Фурье пространственного фильтра. Функцию  $H(u, v)$  принято называть передаточной функцией фильтра.

Основная идея фильтрации в частотной области заключается в подборе передаточной функции фильтра, которая модифицирует  $F(u, v)$  специфическим образом.

Как мы видели в предыдущем примере, результатом низкочастотной фильтрации является размытие (сглаживание) изображения.

Основываясь на теореме о свертке, мы знаем, что для получения соответствующего отфильтрованного изображения в пространственной области нам следует просто вычислить обратное преобразование Фурье от произведения  $H(u, v) \cdot F(u, v)$ .

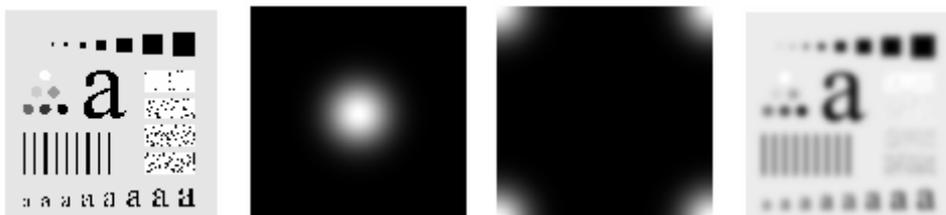
Рассмотрим еще один **пример** использования низкочастотного фильтра. Он реализован нами в виде сценария `ex33.m`.

```
close all;
f=im2double(imread('TestFurier1.tif'));
subplot(2,2,1); imshow(f, []); axis image; % исходное изображение
set(gcf, 'Color', [1 1 1]);
F = fft2(f); % преобразование Фурье
[M, N] = size(f);

% создание центрированного Гауссова фильтра размера M x N
D0=10; % Пороговое значение
H =@(U,V,D0) exp(-(U.^2 + V.^2)/(2*(D0^2))); % Гауссов фильтр
sM=floor(M/2)-floor((M-1)/2);
MM=-floor((M-1)/2)-sM: floor((M-1)/2);
sN=floor(N/2)-floor((N-1)/2);
NN=-floor((N-1)/2)-sN: floor((N-1)/2);
[U,V]=meshgrid(MM,NN);
HH=H(U,V,D0);
subplot(2,2,2); imshow(HH, [ ]); %образ фильтра
axis image;

% децентрирование Гауссова фильтра
HHs=ifftshift(HH);
образ децентрированного фильтра
subplot(2,2,3); imshow(HHs, [ ]); %

G = HHs.*F; % фильтрация в частотной области
g = real(ifft2(G)); % обращение отфильтрованного спектра
subplot(2,2,4); imshow(g, [ ]); % обработанное изображение
% figure, imshow(log(1 + abs(g)), [ ]);
```



Как и ожидалось, отфильтрованное изображение представляет собой размытый образ исходного. Для наглядности фильтр мы привели при

пороговом значении 50, а обработанное изображение показано при пороговом значении 10.

Здесь через  $f$  обозначено исходное изображение, а через  $g$  — результат фильтрации. Передаточная функция  $H(u, v)$  должна иметь те же размеры, что и исходное изображение. Если функция  $H(u, v)$  была вещественной (здесь мы рассматриваем только такие фильтры), то фазовая часть произведения не меняется, что видно из фазового уравнения, так как при умножении вещественной и мнимой части комплексного числа на одно и то же вещественное число фазовый угол не меняется. Такие фильтры принято называть фильтрами с нулевым сдвигом фазы. В нашем примере передаточная функция децентрированного фильтра Гаусса обозначена через  $H_h(u, v)$  и умножается на вещественную и мнимую части  $F(u, v)$ . Так, что мы имеем дело с фильтром с нулевым сдвигом фазы.

Попробуйте использовать центрированный фильтр, заменив строку  $G = H_h .* F$ ; на  $G = H .* F$ ;

Таким образом, можно выполнять одну и ту же фильтрацию, как в пространственной, так и в частотной области. В принципе полезно знать, как конвертировать пространственные фильтры в эквивалентную частотную форму, чтобы иметь возможность сравнивать эти два подхода. Самый очевидный метод построения частотного фильтра  $H$ , отвечающего пространственному фильтру  $h$ , состоит в выполнении команды  $H = \text{fft2}(h, M, N)$ , где значения вектора  $[M, N]$  зависят от размеров изображения, которое будет фильтроваться. Фактически, задание  $M$  и  $N$  расширяет исходное изображение нулями, а затем строится ДПФ расширенного изображения.

Функция `imfilter` использует корреляцию. При этом начало координат фильтра расположено в его центре. Поэтому для достижения эквивалентных результатов при Фурье подходе необходимо выполнить определенную предварительную обработку данных. В ИРТ пакете имеется функция `freqz2`, которая совершает необходимые преобразования и ее выходом служит соответствующий фильтр в частотной области. Интересующая нас команда выглядит следующим образом:

$$H = \text{freqz2}(h, R, C),$$

где  $h$  — это двумерный пространственный фильтр, а  $H$  — соответствующий ему двумерный фильтр в частотной области. Здесь  $R$  обозначает число строк, а  $C$  — число столбцов фильтра  $H$ . Если записать функцию `freqz2` без обозначения выходной переменной, то абсолютная величина  $H$  будет изображена в виде трехмерного аксонометрического графика.

В противоположность низкочастотной фильтрации, которая приводит к размытию изображений, высокочастотная фильтрация повышает резкость изображения, ослабляя низкие частоты и оставляя высокие частоты преобразования Фурье относительно неизменными.

Имея передаточную функцию  $H_{lp}(u,v)$  низкочастотного фильтра, можно получить передаточную функцию соответствующего высокочастотного фильтра с помощью формулы

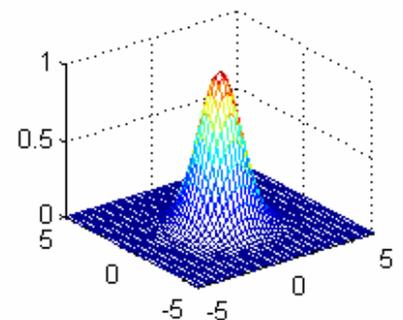
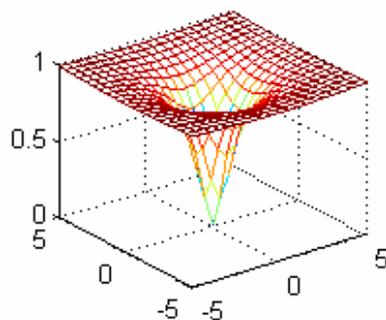
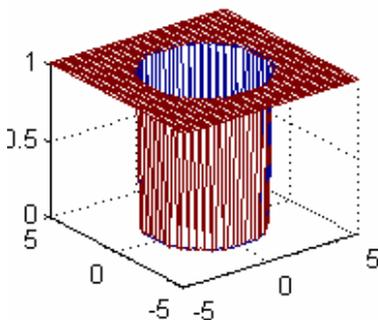
$$H_{hp}(u,v) = 1 - H_{lp}(u,v).$$

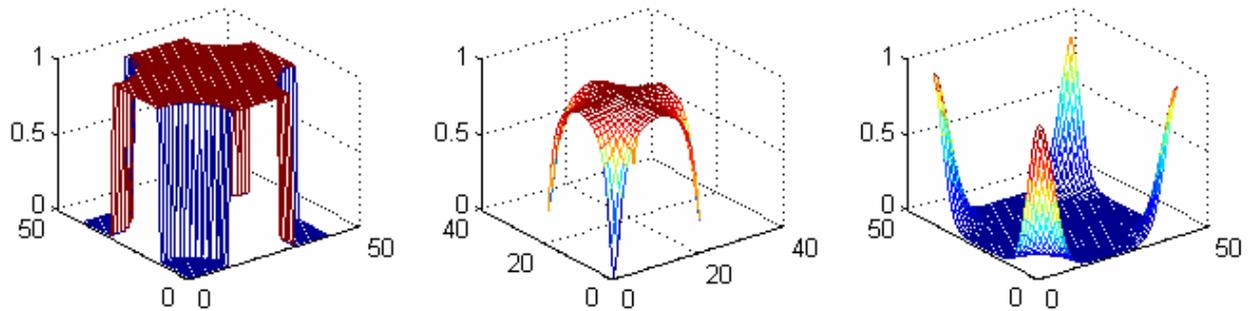
**Пример.** На следующем рисунке приведены изображения идеального высокочастотного фильтра, а также фильтров Баттерворта и Гаусса высокой частоты. Они построены с помощью следующих команд (ex34.m)

```
% Построение высокочастотных фильтров
close all;
% Идеальный высокочастотный фильтр (левые изображения)
HEh =@(U,V,D0) (1-1.*((U.^2 + V.^2)<D0^2));
[U,V]=meshgrid(-5:0.25:5);HHEh=HEh(U,V,3);
subplot(2,3,1);mesh(U,V,HHEh);
% построение нецентрированного фильтра
HHsh=ifftshift(HHEh);
subplot(2,3,4); mesh(HHsh);

% высокочастотный фильтр Баттерворта (изображения в центре)
HBh =@(U,V,D0,n) 1-1./(1+((U.^2 + V.^2)./D0^2).^n);
[U,V]=meshgrid(-5:0.5:5);HHBh=HBh(U,V,1,1);
subplot(2,3,2); mesh(U,V,HHBh);
% построение нецентрированного фильтра
HHsh=ifftshift(HHBh);
subplot(2,3,5); mesh(HHsh);

% высокочастотный фильтр Гаусса (изображения справа)
Hh =@(U,V,D0) 1-exp(-(U.^2 + V.^2)/(2*(D0^2)));
[U,V]=meshgrid(-5:0.25:5);HHh=H(U,V,1);
subplot(2,3,3); mesh(U,V,HHh);
% построение нецентрированного фильтра
HHsh=ifftshift(HHh);
subplot(2,3,6); mesh(HHsh);
set(gcf,'Color',[1 1 1]);
```





Так же как и при низкочастотной фильтрации для этих фильтров нужно подбирать размеры и выполнять их децентрирование.

**Пример.** Высокочастотная фильтрация (ex35.m). Этот сценарий отличается от ex33.m только изменением функции фильтра.

```
% Пример высокочастотной фильтрации
close all;
f=im2double(imread('TestFurier1.tif'));
subplot(2,2,1); imshow(f,[]); axis image; % исходное изображение
set(gcf,'Color',[1 1 1]);

F = fft2(f); % преобразование Фурье
[M, N] = size(f);

% создание центрированного высокочастотного Гауссова фильтра
D0=10; % Пороговое значение
Nh=@(U,V,D0) 1-exp(-(U.^2 + V.^2)/(2*(D0^2))); %Гауссов ВЧ фильтр
sM=floor(M/2)-floor((M-1)/2);
MM=-floor((M-1)/2)-sM: floor((M-1)/2);
sN=floor(N/2)-floor((N-1)/2);
NN=-floor((N-1)/2)-sN: floor((N-1)/2);
[U,V]=meshgrid(MM,NN);
HHh=Nh(U,V,D0);
subplot(2,2,2);imshow(HHh,[]); %образ фильтр
axis image;

% децентрирование Гауссова фильтра
HHsh=ifftshift(HHh);
% образ децентрированного фильтра
subplot(2,2,3);imshow(HHsh,[]);

G = HHsh.*F; % фильтрация в частотной области
g = real(ifft2(G)); % обращение отфильтрованного спектра
subplot(2,2,4);imshow(g,[]); % отфильтрованное изображение
```



Справа приведено обработанное изображение. Чем больше  $D_0$ , тем меньше низких частот остается в изображении. Заметно повышение четкости границ и усиление других контуров. Вместе с тем, поскольку средняя яркость изображения задается величиной  $F(0,0)$ , а, как мы знаем, высокочастотная фильтрация гасит низкие частоты, то отфильтрованное изображение потеряло большую часть фоновой тональности, имевшуюся на исходном изображении. Эта проблема рассматривается в следующем примере.

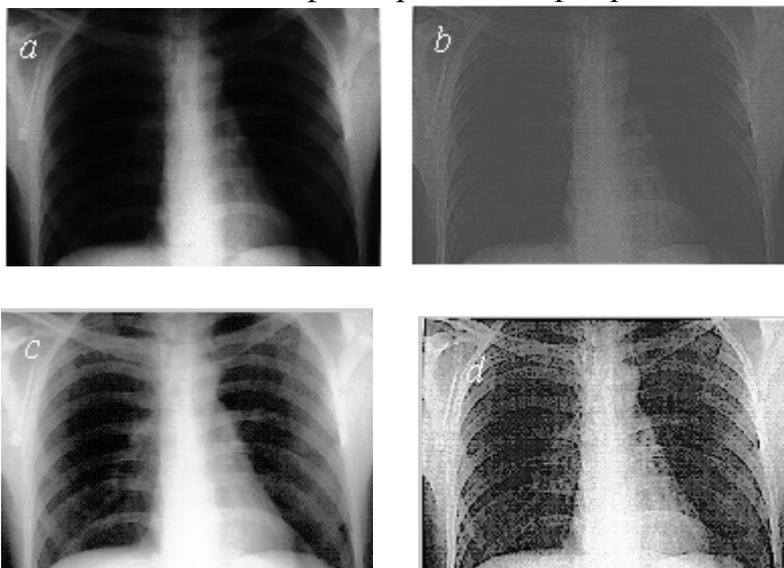
Как отмечено выше, высокочастотные фильтры значительно уменьшают среднюю яркость изображения, отбрасывая коэффициент  $F(0,0)$ , в итоге среднее становится равным нулю. Чтобы компенсировать этот недостаток, необходимо добавить некоторое смещение к высокочастотному фильтру. Если смещение комбинируется вместе с умножением фильтра на коэффициент усиления, больший 1, то такая процедура называется фильтрацией с усилением высоких частот. Поскольку этот коэффициент больше единицы, то в результате происходит увеличение высоких частот. Конечно, амплитуда низких частот при этом тоже увеличивается, однако этот эффект меньше влияет на низкие частоты, поскольку смещение мало по сравнению с коэффициентом усиления. Высокочастотное усиление имеет передаточную функцию

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

где  $a$  — это смещение,  $b$  — коэффициент усиления, а  $H_{hp}(u, v)$  — передаточная функция фильтра высоких частот.

**Пример.** Комбинированная фильтрация с усилением высоких частот и гистограммная эквализация.

На следующем рисунке показано: *a*) исходное изображение, *b*) обработанное высокочастотным фильтром Баттерворта, *c*) исходное изображение после гистограммной эквализации, *d*) обработанное высокочастотным фильтром Баттерворта после гистограммной эквализации.



Эти изображения получены после выполнения следующего сценария `ex36.m`

```

% Пример высокочастотной фильтрации совместно
% с гистограммной эквализацией
close all;
f=im2double(imread('Xray.tif'));
subplot(2,2,1); imshow(f,[]); % исходное изображение
axis image;
set(gcf,'Color',[1 1 1]);

F = fft2(f); % преобразование Фурье
[M, N] = size(f);

% создание центрированного высокочастотного фильтра
% Баттерворта размера M x N
D0=25; % Пороговое значение
NBh=@(U,V,D0,n) 1-1./(1+((U.^2 + V.^2)./D0^2).^n);
sM=floor(M/2)-floor((M-1)/2);
MM=-floor((M-1)/2)-sM: floor((M-1)/2);
sN=floor(N/2)-floor((N-1)/2);
NN=-floor((N-1)/2)-sN: floor((N-1)/2);
[U,V]=meshgrid(MM,NN);
NNh=NBh(U',V',D0,2); % требуется поворот фильтра
subplot(2,2,2);imshow(NNh,[]); %центрированный фильтр
axis image;

% децентрирование фильтра Баттерворта
NNsh=ifftshift(NNh);
% образ децентрированного фильтра
subplot(2,2,3);imshow(NNsh,[]);
H = 0.5 + 2*NNsh; % усиление частот

G = H.*F; % фильтрация в частотной области
g = real(ifft2(G)); % обращение отфильтрованного образа
subplot(2,2,4);imshow(g,[]); % отфильтрованное изображение
axis image;

% гистограммная эквализация исходного изображения
fhs = histeq(f, 256);
figure, imshow(fhs);
% гистограммная эквализация отфильтрованного изображения
ghe = histeq(g, 256);
figure, imshow(ghe);

```

Выше представлен рентгеновский снимок грудной клетки  $f$ . Поскольку рентгеновские лучи хуже поддаются фокусировке по сравнению со световыми лучами, рентгеновские снимки, как правило, выглядят расплывчато. Цель этого примера заключается в повышении резкости данного изображения. В этом конкретном примере яркость изображения сдвинута в темную область градационной шкалы, поэтому мы воспользуемся этим обстоятельством для демонстрации пространственной обработки, которая удачно дополняет фильтрацию в частотной области.

На рисунке *b*) изображен результат высокочастотной фильтрации изображения *a*) с использованием высокочастотного фильтра Баттерворта порядка 2. Высокочастотная фильтрация не очень сильно зависит от параметра  $D_0$ . Как и ожидалось, результат фильтрации не очень выразителен, однако на нем слабо различимы основные контуры изображения. Рисунок *d*) демонстрирует преимущество фильтрации с усилением высоких частот (в данном случае,  $a = 0.5$  и  $b = 2.0$ ), при котором сохранен общий яркостной тон исходного изображения.

Как отмечалось ранее, изображение, характеризующееся узким диапазоном яркости, является идеальным кандидатом для гистограммной эквализации. Из рис *d*) видно, что этот метод действительно продолжает улучшать изображение. Обратите внимание на четкое выявление структур костей и других деталей, которых даже не видно на трех предыдущих изображениях. В последнем изображении можно обнаружить некоторое усиление шумов, но это явление весьма характерно для рентгеновских снимков при растяжении диапазона их яркости. Результат, полученный при совместном использовании фильтрации с усилением высоких частот и метода гистограммной эквализации, превосходит любой результат, который можно было бы получить, применяя каждый из этих методов по отдельности.