

# Сжатие изображений и преобразование Фурье

Курс «Компьютерное зрение»

30 сентября 2025 года

Тимур Мамедов



# Что такое сжатие изображений?



**Сжатие изображений** — это процесс уменьшения объема данных, необходимых для хранения или передачи изображений, достигаемый за счет удаления избыточной или незначительной информации

## **Основные типы сжатия:**

1. Сжатие без потерь: сохраняет оригинальное качество изображения (пример: PNG)
2. Сжатие с потерями: уменьшает размер файла, удаляя некоторые данные, что может привести к ухудшению качества (пример: JPEG)

# PNG



Формат PNG появился в 1995 году в качестве замены GIF

## Основные характеристики PNG:

- Сжатие без потерь
- Поддержка альфа-канала
- Наличие цветовых палитр
- Высокое качество изображений
- Широкая совместимость и простая имплементация

# PNG: как устроен



PNG использует алгоритм сжатия без потерь DEFLATE, комбинирующий метод Хаффмана и LZ77, для уменьшения размера файла. PNG-файл состоит из различных «блоков» (чанков), каждый из которых несет свою функцию

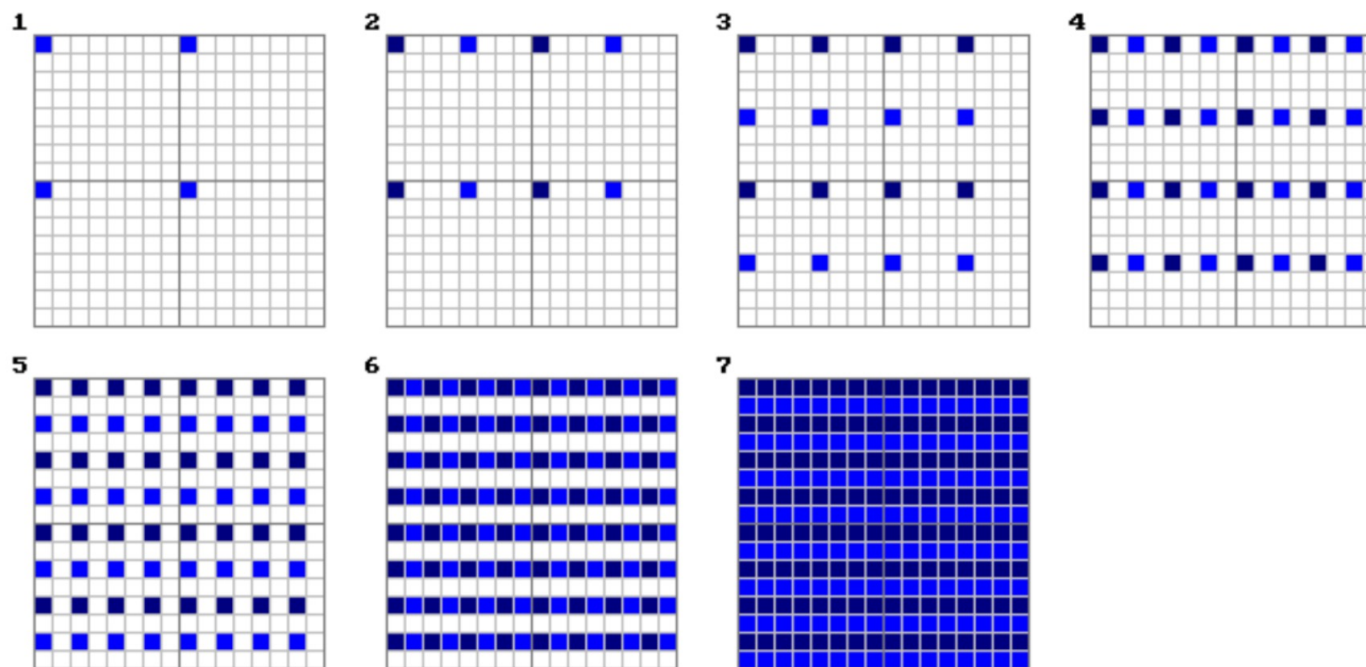
## Основные блоки:

- IHDR: содержит информацию о размере изображения, глубине цвета и других параметрах
- PLTE: определяет индексированную палитру цветов, используемых в картинке
- IDAT: содержит сжатые данные изображения. Картинка может быть разбита на несколько IDAT-чанков для потоковой передачи
- IEND: обозначает конец файла

# PNG: потоковая передача



PNG-изображения могут передаваться по сети по частям с помощью алгоритма Adam7



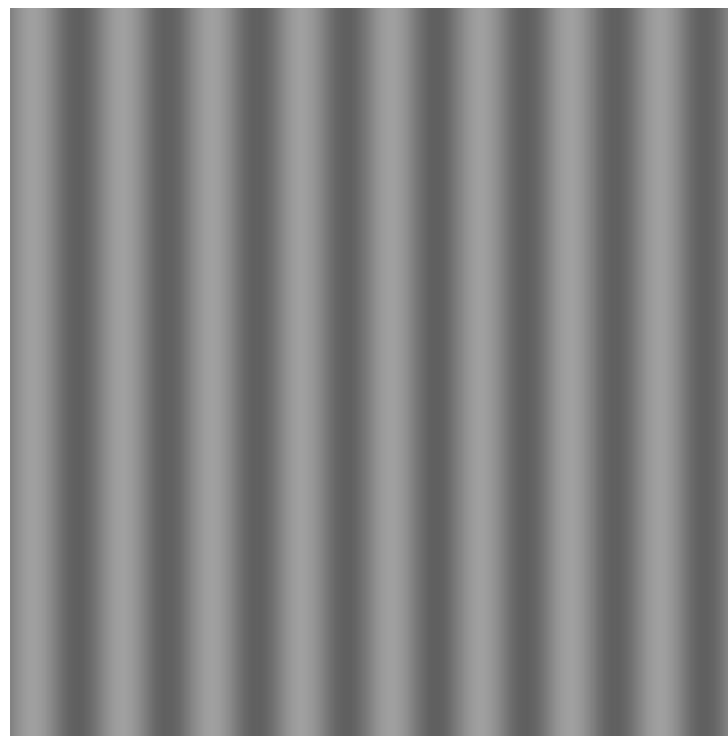
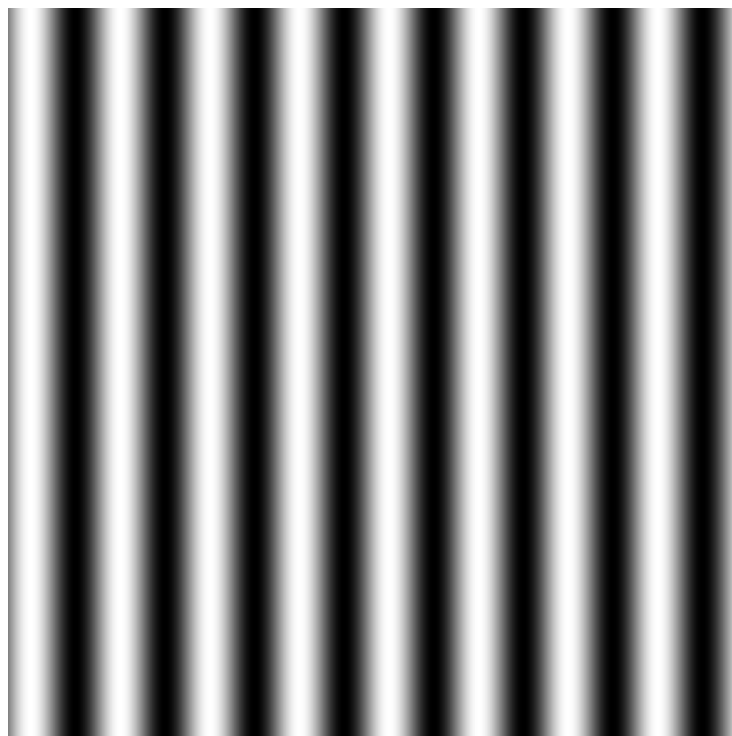
# JPEG



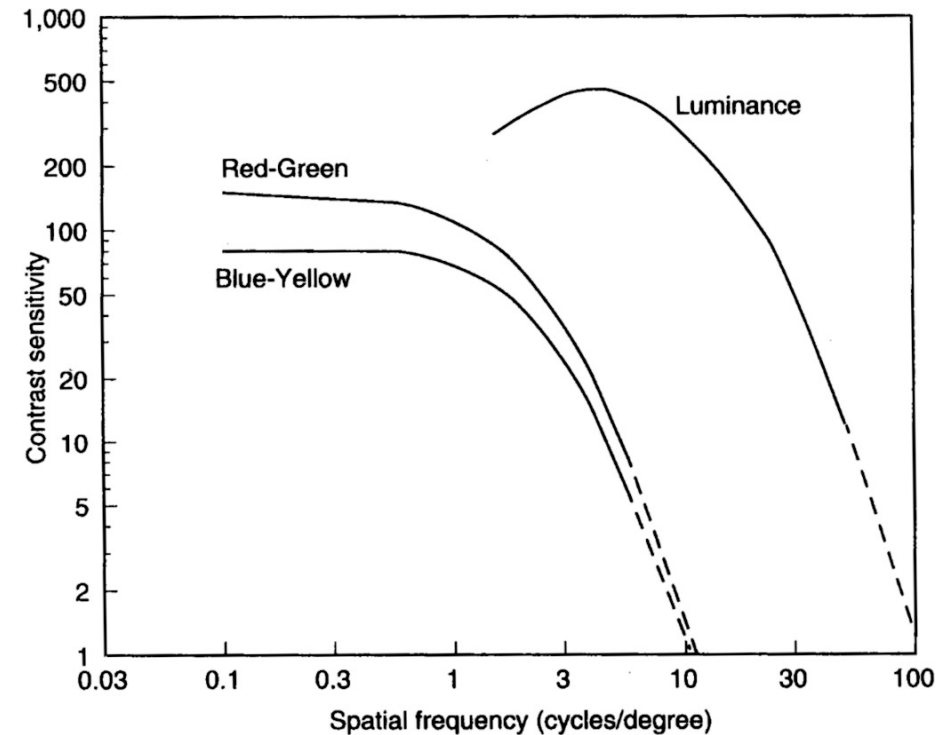
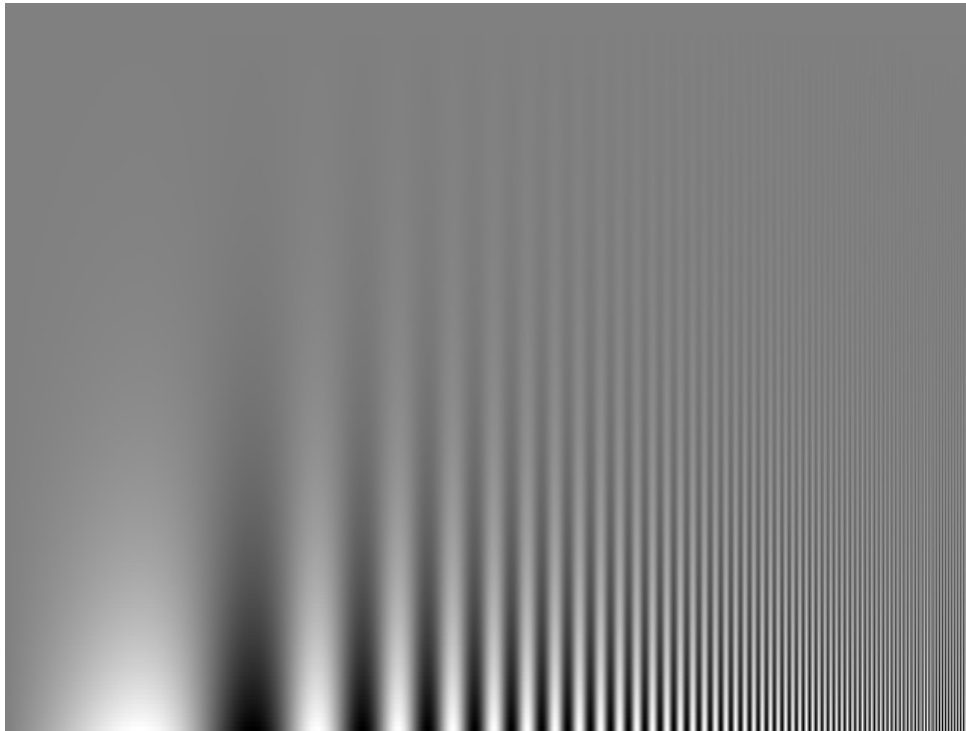
JPEG (Joint Photographic Experts Group) является одним из самых популярных форматов сжатия изображений с потерями, особенно для фотографий с большим количеством цветов

**Так как данный формат подразумевает сжатие картинки с потерями, то при его создании учитывались особенности человеческого восприятия изображений**

# JPEG: восприятие изображений



# JPEG: восприятие изображений



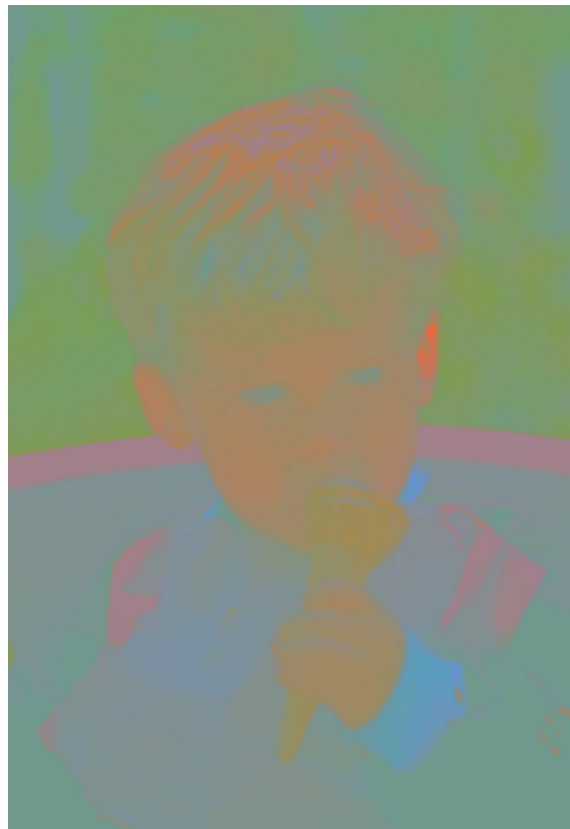
Оказалось, что люди наиболее восприимчивы к яркости изображений => данный факт стоит учитывать при сжатии картинок



# JPEG: восприятие изображений



RGB



Цвет



Яркость

# JPEG: восприятие изображений



RGB

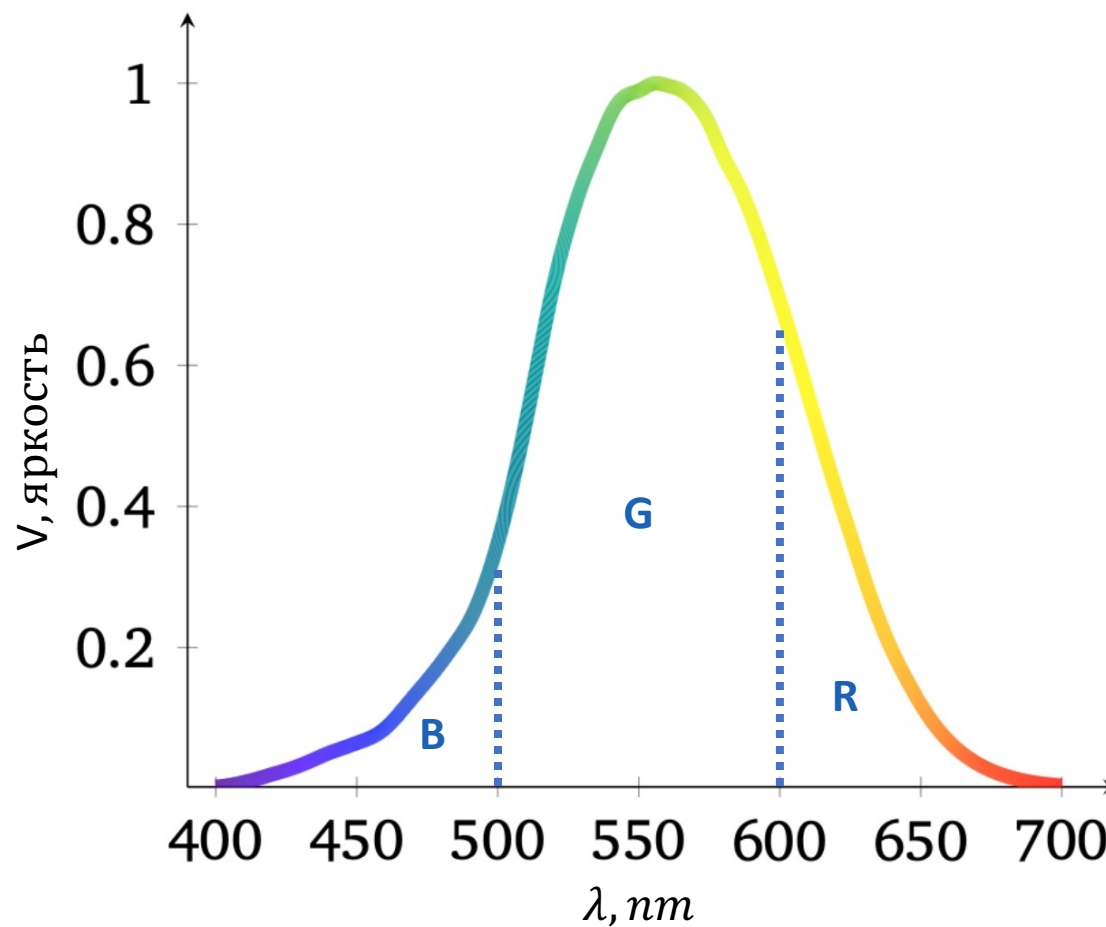


Размытый цвет



Размытая яркость

# JPEG: чувствительность глаза к яркости





# JPEG: переход из RGB в YCbCr



$$RGB \rightarrow Y C_b C_r$$

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B$$

$$C_b = -0.169 \cdot R - 0.331 \cdot G + 0.500 \cdot B + 128$$

$$C_r = 0.500 \cdot R - 0.419 \cdot G - 0.081 \cdot B + 128$$



# JPEG: уменьшение цветовых компонент



Все дальнейшие шаги в JPEG выполняются независимо для каждого канала

# JPEG: деление на блоки и сдвиг на 128



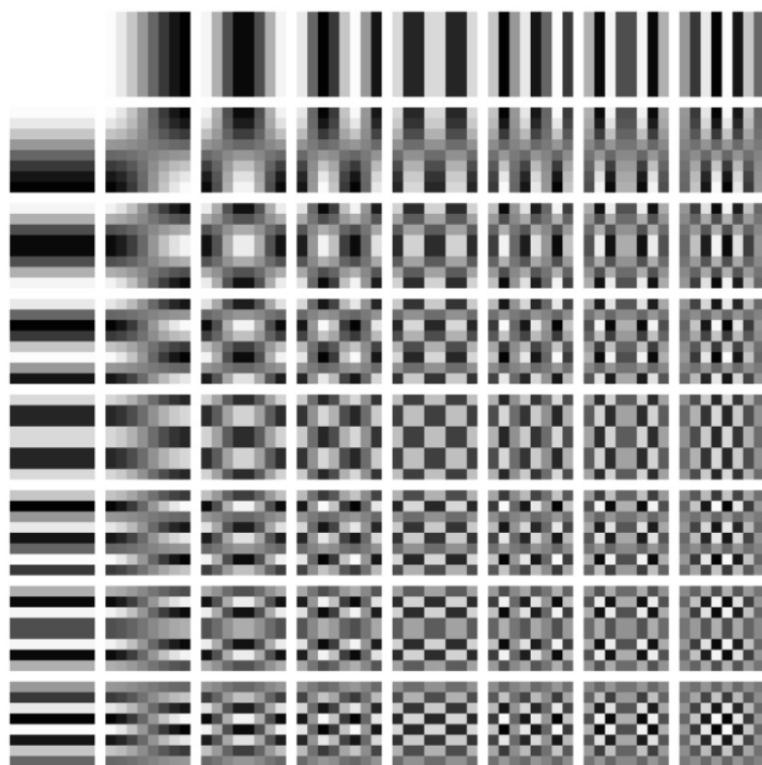
Все каналы делятся на блоки 8x8 и происходит сдвиг значений на 128:

139	144	149	153	155	155	155	155	11	16	21	25	27	27	27	27
144	151	153	156	159	156	156	156	16	23	25	28	31	28	28	28
150	155	160	163	158	156	156	156	22	27	32	35	30	28	28	28
159	161	162	160	160	159	159	159	31	33	34	32	32	31	31	31
159	160	161	162	162	155	155	155	31	32	33	34	34	27	27	27
161	161	161	161	160	157	157	157	33	33	33	33	32	29	29	29
162	162	161	163	162	157	157	157	34	34	33	35	34	29	29	29
162	162	162	161	163	158	158	158	34	34	34	33	35	30	30	30

Блок

Сдвинутый блок

# JPEG: применение ДКП



$$\begin{array}{c} x \\ \longrightarrow \end{array}
 \begin{bmatrix}
 11 & 16 & 21 & 25 & 27 & 27 & 27 & 27 \\
 16 & 23 & 25 & 28 & 31 & 28 & 28 & 28 \\
 22 & 27 & 32 & 35 & 30 & 28 & 28 & 28 \\
 31 & 33 & 34 & 32 & 32 & 31 & 31 & 31 \\
 31 & 32 & 33 & 34 & 34 & 27 & 27 & 27 \\
 33 & 33 & 33 & 33 & 32 & 29 & 29 & 29 \\
 34 & 34 & 33 & 35 & 34 & 29 & 29 & 29 \\
 34 & 34 & 34 & 33 & 35 & 30 & 30 & 30
 \end{bmatrix}
 \begin{array}{c} \downarrow y \end{array}$$

$$\begin{array}{c} u \\ \longrightarrow \end{array}
 \begin{bmatrix}
 235.6 & -1.0 & -12.1 & -5.2 & 2.1 & -1.7 & -2.7 & 1.3 \\
 -22.6 & -17.5 & -6.2 & -3.2 & -2.9 & -0.1 & 0.4 & -1.2 \\
 -10.9 & -9.3 & -1.6 & 1.5 & 0.2 & -0.9 & -0.6 & -0.1 \\
 -7.1 & -1.9 & 0.2 & 1.5 & 0.9 & -0.1 & 0.0 & 0.3 \\
 -0.6 & -0.8 & 1.5 & 1.6 & -0.1 & -0.7 & 0.6 & 1.3 \\
 1.8 & -0.2 & 1.6 & -0.3 & -0.8 & 1.5 & 1.0 & -1.0 \\
 -1.3 & -0.4 & -0.3 & -1.5 & -0.5 & 1.7 & 1.1 & -0.8 \\
 -2.6 & 1.6 & -3.8 & -1.8 & 1.9 & 1.2 & -0.6 & -0.4
 \end{bmatrix}
 \begin{array}{c} \downarrow v \end{array}$$

Амплитуды



# JPEG: квантование



**Мысль:** человеческий глаз хорошо улавливает низкие частоты, а высокие – намного хуже. Следовательно, можно удалить высокие частоты и не очень сильно испортить картинку, но при этом существенно уменьшить ее объем

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Матрица квантования

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Амплитуды после квантования



# JPEG: квантование



Матрицы квантования были получены после анализа статистик на реальных изображениях, а также модели человеческого восприятия картинок

Помимо всего прочего, данные матрицы могут зависеть от выбранного пользователем значения параметра Quality Factor  $Q \in [0,100]$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Матрица квантования яркости

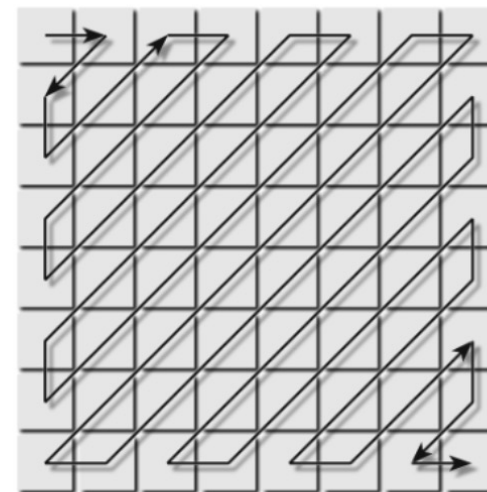
17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Матрица квантования цвета

# JPEG: зигзаг-сканирование и сжатие



$$\begin{bmatrix} 15 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



15 0 -2 -1 -1 -1 0 0 -1 0 ...0

Далее данная последовательность сжимается с помощью Zero Run-Length Encoding

# JPEG: итоговый алгоритм



1. Изображение переводится из RGB в YCbCr
2. Уменьшаются цветовые компоненты
3. Все каналы делятся на блоки 8x8
4. Все элементы блоков переводятся из  $[0, 255]$  в  $[-128, 127]$
5. Ко всем блокам последовательно применяются:
  - Дискретное косинусное преобразование
  - Квантование
  - Зигзаг-сканирование
  - Сжатие

# Преобразование Фурье



$$G(\omega) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} g(x) e^{-i\omega x} dx$$

Diagram labels for the Fourier transform equation:

- спектр** (spectrum) points to  $G(\omega)$
- сигнал** (signal) points to  $g(x)$
- базис** (basis) points to  $e^{-i\omega x}$

Прямое преобразование Фурье

$$g(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} G(\omega) e^{i\omega x} d\omega$$

Обратное преобразование Фурье

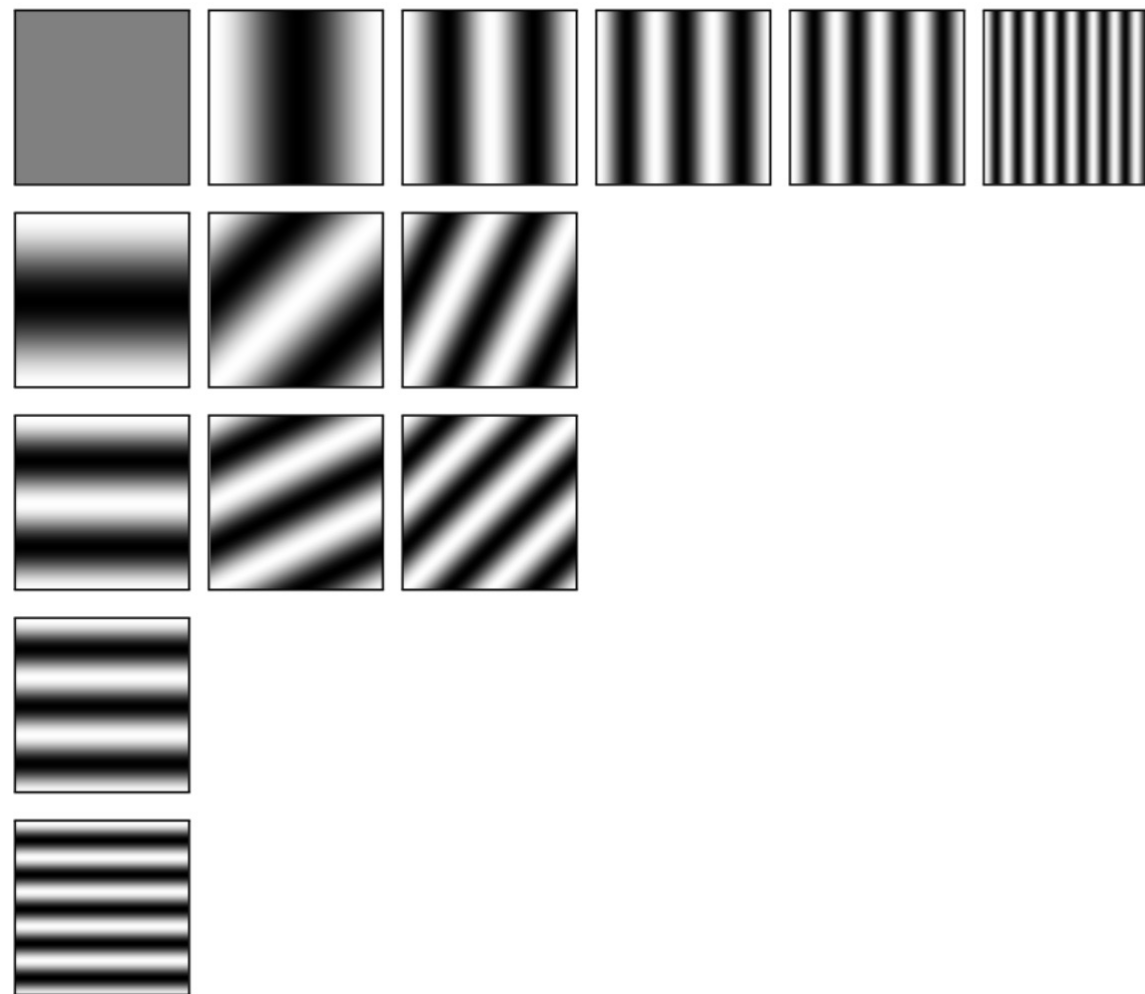
# Дискретное преобразование Фурье



$$G(m) = \frac{1}{\sqrt{M}} \sum_{u=0}^{M-1} g(u) e^{-i2\pi \frac{mu}{M}}, \quad 0 \leq m < M$$

$$g(u) = \frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} G(m) e^{i2\pi \frac{mu}{M}}, \quad 0 \leq u < M$$

# Базис Фурье для изображений

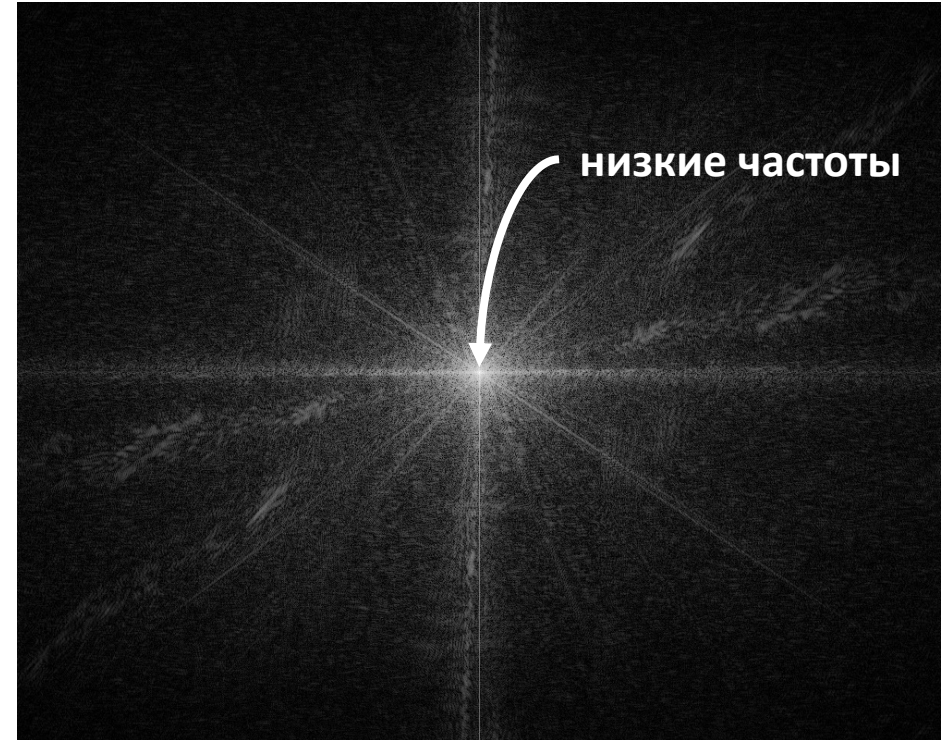




# Преобразование Фурье



Пространственный домен



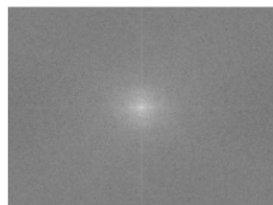
Частотный домен

```
freq = log(1 + abs(fftshift(fft2(img))))
```

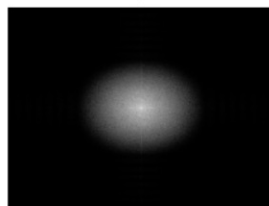
# Теорема о свертке



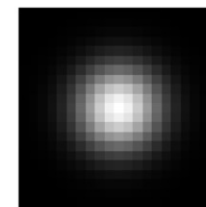
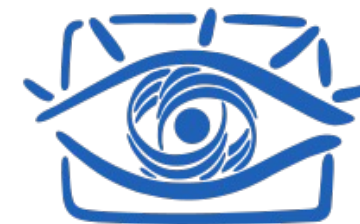
$\downarrow \mathcal{F}$



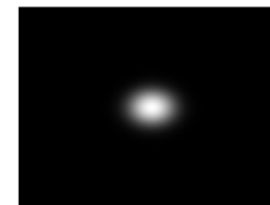
$*$



$\downarrow \mathcal{F}^{-1}$



$\downarrow \mathcal{F}$



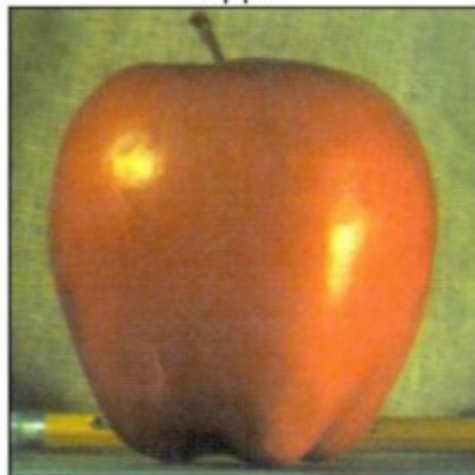
предварительно  
увеличиваем картинку



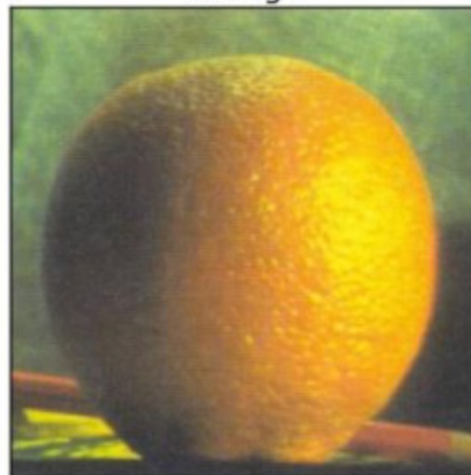
# Склейка изображений



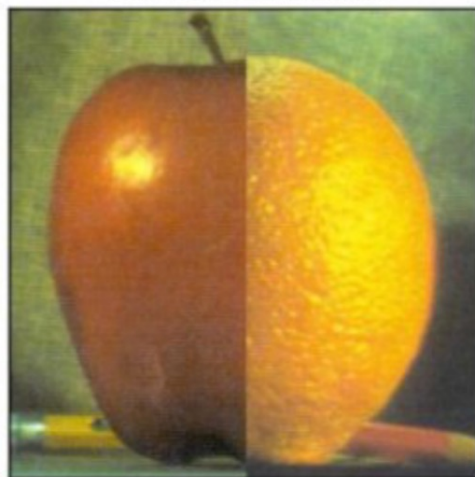
Apple



Orange



Direct Connection



Pyramid Blending



# Склейка изображений



## Вход:

Изображения  $A$ ,  $B$  и маска  $M$

## Алгоритм:

1. Для изображений  $A$  и  $B$  вычисляются лапласовские пирамиды  $LA$  и  $LB$
2. Для маски  $M$  вычисляется гауссовская пирамида  $GM$
3. Лапласовские пирамиды смешиваются:  $LS = GM * LA + (1 - GM) * LB$
4. Конечный результат вычисляется из  $LS$  суммированием

# Резюме



## На этой лекции мы рассмотрели:

- Методы сжатия изображений без потерь (PNG) и с потерями (JPEG)
- Все этапы алгоритма JPEG
- Преобразование Фурье – инструмент для анализа изображений и фильтров
- Теорему о свертке, позволяющую эффективнее применять большие фильтры
- Алгоритм склейки изображений с помощью лапласовской пирамиды