

ITMO
More than a
UNIVERSITY

Современная теория информации

Лекция 8. Сжатие изображений. Часть 1.

Содержание лекции

- ① Цифро-аналоговое преобразование изображений.
- ② Основные принципы сжатия изображений.
- ③ Преобразование цветового пространства.
- ④ Кодирование с предсказанием.
- ⑤ Кодирование с преобразованием.
- ⑥ Алгоритм JPEG

Цифро-аналоговое преобразование изображений

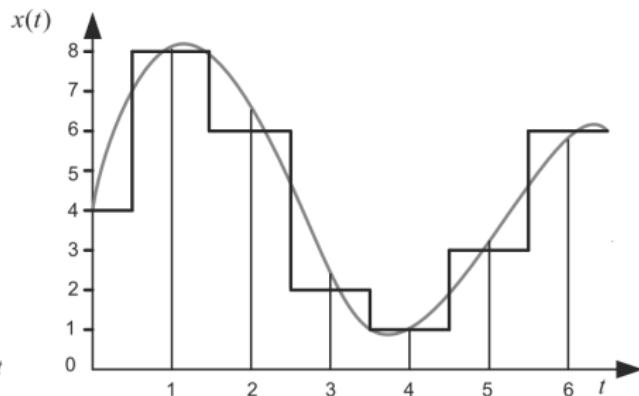
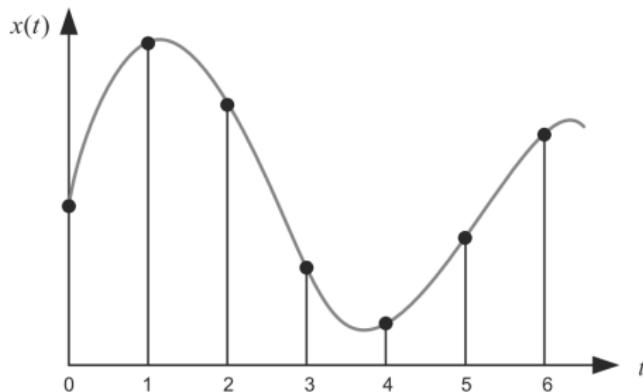
Аналоговые и цифровые сигналы

Под **сигналом** понимается физический процесс (например, изменяющееся во времени напряжение), отображающий некоторую информацию или сообщение.

- Аналоговые сигналы - это непрерывные функции непрерывного аргумента, например такого как время и/или пространство.
- Дискретные сигналы могут быть дискретными по множеству значений функции или по множеству значений аргумента.
Например, если значения аналогового сигнала взяты через определенные интервалы времени, то такой сигнал называется **дискретным по времени**;
- Если у дискретизованного по времени сигнала отсчеты также принимают значения из некоторого дискретного множества значений, то такой сигнал называется **цифровым**.

Цифро-аналоговое преобразование изображений

Дискретизация по времени и по уровню

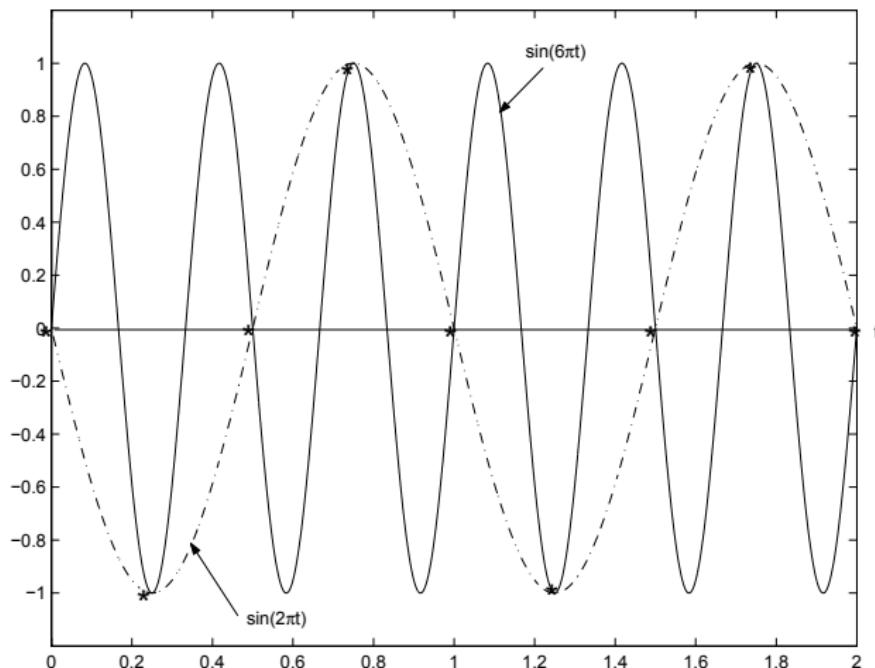


Для того, чтобы поместить аналоговый (непрерывный сигнал) в цифровое устройство необходимо преобразовать его в цифровой сигнал. Аналого-цифровое преобразование (ЦАП) включает в себя две основные операции:

- Операция дискретизации или взятия отсчетов непрерывного сигнала (квантование по времени);
- Операция квантования непрерывного сигнала по уровню.

Цифро-аналоговое преобразование изображений

Выбор частоты дискретизации по времени



Как правильно выбрать интервал дискретизации по времени?

Цифро-аналоговое преобразование изображений

Спектр непрерывного сигнала

Пусть $x(t)$ – непрерывная одномерная функция. Преобразование Фурье (спектр) для $x(t)$ определяется следующим образом:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt,$$

где $\omega = 2\pi f$ – круговая частота. Эта комплексная функция может быть представлена в виде

$$X(f) = A(f)e^{j\varphi(f)},$$

где $|A(f)|$ называется амплитудным спектром сигнала или амплитудно-частотной характеристикой сигнала, а $\varphi(f)$ называется фазовым спектром или фазово-частотной характеристикой.

Восстановить сигнал по его спектру можно при помощи обратного преобразования Фурье:

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{-j\omega t} df.$$

Цифро-аналоговое преобразование изображений

Теорема Котельникова-Найквиста

Пусть задана функция $x(t)$ и её спектр имеет вид:

$$\begin{cases} X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} dt, & \text{если } |f| \leq f_0. \\ X(f) = 0, & \text{если } |f| > f_0. \end{cases}$$

Тогда эта функция полностью определяется своими мгновенными значениями в моменты, отстоящие друг от друга на $\frac{1}{2f_0}$ секунд:

$$x(t) = \sum_k x\left(\frac{k}{2f_0}\right) \frac{\sin\pi(2f_0t - k)}{\pi(2f_0t - k)}.$$

Цифро-аналоговое преобразование изображений

Пространственный спектр изображения

Спектр $L(\omega_x, \omega_y)$ изображения, описываемого функцией яркости $I(x, y)$, определяется следующим образом:

$$L(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) e^{-j(\omega_x x + \omega_y y)} dx dy,$$

где ω_x и ω_y – круговые пространственные частоты спектра в направлении осей x и y .

Спектральная интенсивность изображения:

$$S(\omega_x, \omega_y) = \frac{1}{x_0 y_0} |L(\omega_x, \omega_y)|^2,$$

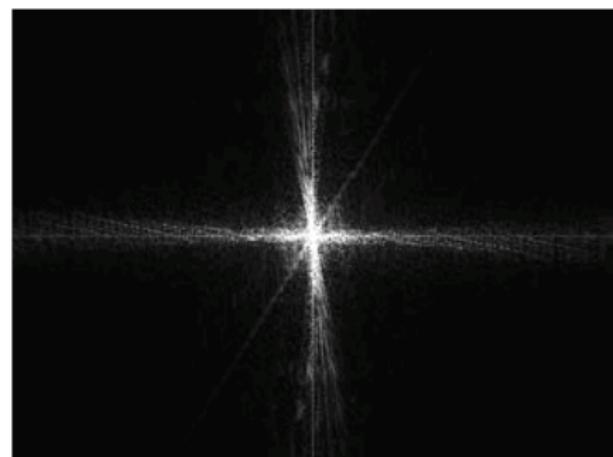
где $x_0 y_0$ – площадь прямоугольника, в которое вписано изображение.

Восстановить функцию яркости по его спектру можно при помощи обратного двумерного преобразования Фурье:

$$I(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} L(\omega_x, \omega_y) e^{j(\omega_x x + \omega_y y)} d\omega_x d\omega_y.$$

Цифро-аналоговое преобразование изображений

Пространственный спектр изображения



Реальные изображения в основном состоят из вертикальных или горизонтальных объектов.

Цифро-аналоговое преобразование изображений

Теорема Котельникова-Найквиста для изображения

Пусть задана функция яркости изображения $I(x, y)$ со спектром

$$\begin{cases} L(\omega_x, \omega_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) e^{-j(2\pi f_x x + 2\pi f_y y)} dx dy, |f_x| \leq f_x^0, |f_y| \leq f_y^0. \\ L(\omega_x, \omega_y) = 0, \text{ иначе.} \end{cases}$$

Тогда эта функция полностью определяется своими мгновенными значениями в моменты, отстоящие друг от друга на интервалы

$\Delta_x = \frac{1}{2f_x^0}$ и $\Delta_y = \frac{1}{2f_y^0}$ в направлении осей x и y :

$$I(x, y) = \sum_n \sum_k L(n\Delta_x, k\Delta_y) \frac{\sin 2\pi f_x^0 (x - n\Delta_x)}{2\pi f_x^0 (x - n\Delta_x)} \cdot \frac{\sin 2\pi f_y^0 (y - k\Delta_y)}{2\pi f_y^0 (y - k\Delta_y)}$$

Цифро-аналоговое преобразование изображений

Пример нарушения теоремы (aliasing)



Цифро-аналоговое преобразование изображений

Равномерное скалярное квантование

Наиболее простой способ аппроксимации последовательности сообщений $\mathbf{x} = \{x_1, \dots, x_n\}$ может быть реализован при помощи процедуры равномерного скалярного квантования, которая каждому символу x_i сопоставляет номер кванта

$$z_i = \text{sign}(x_i) \left\lfloor \frac{|x_i| + \Delta/2}{\Delta} \right\rfloor,$$

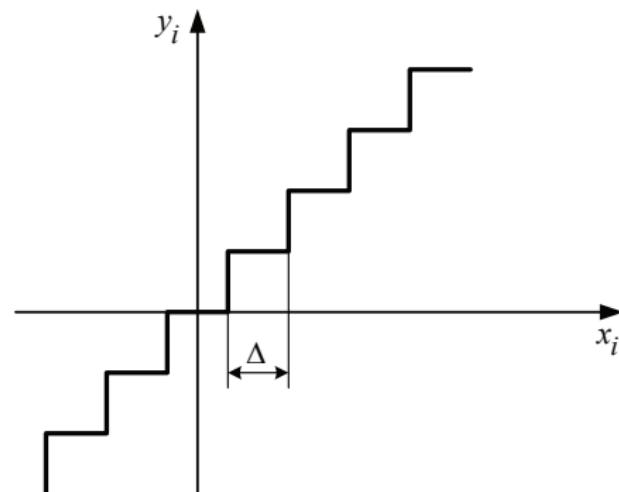
где Δ - шаг квантования, $\lfloor x \rfloor$ - означает операцию округления до ближайшего целого, не превышающего x , $\text{sign}(x_i) = -1$, если $x_i < 0$, иначе $\text{sign}(x_i) = 1$.

При этом, аппроксимирующее множество $\mathbf{y} = \{y_1, \dots, y_n\}$ вычисляется как

$$y_i = \Delta \cdot z_i.$$

Цифро-аналоговое преобразование изображений

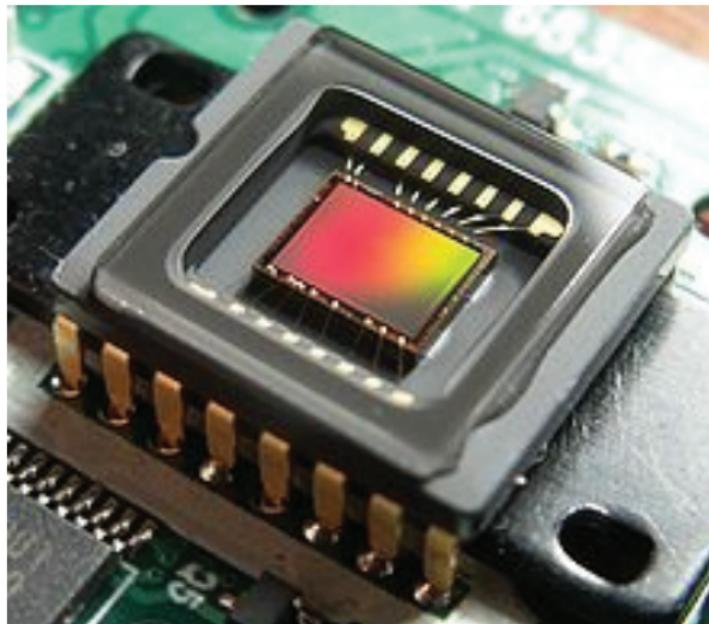
Равномерное скалярное квантование



$$\text{Средняя ошибка квантования } \varepsilon = \int_{-\infty}^{\infty} (x_i - y_i)^2 f(x_i) dx \approx \frac{\Delta^2}{12}.$$

Цифро-аналоговое преобразование изображений

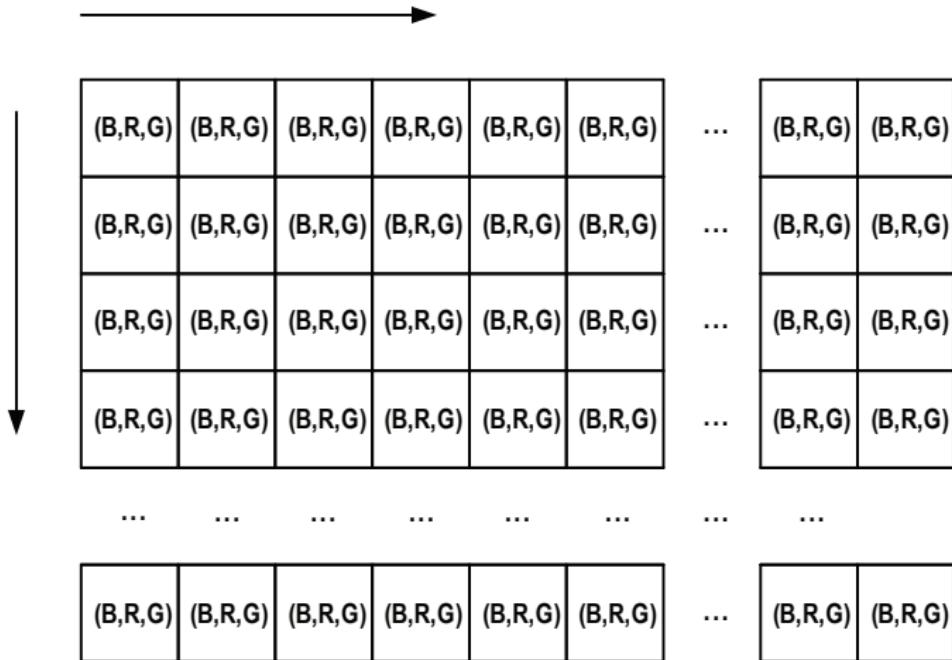
Цифро-аналоговая интегральная микросхема



Светочувствительная матрица — цифро-аналоговая интегральная микросхема, состоящая из фотодиодов (светочувствительных элементов).

Цифро-аналоговое преобразование изображений

Цифровое представление изображений в формате RGB24



$R, G, B \in [0, \dots, 255]$ or 24 бит на пиксель.

Основные принципы сжатия изображений

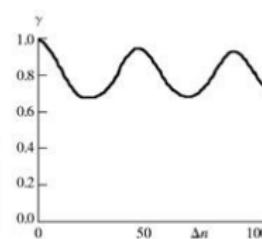
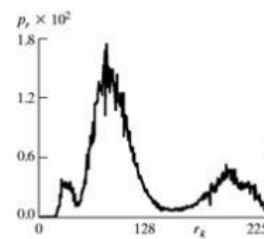
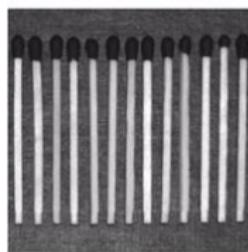
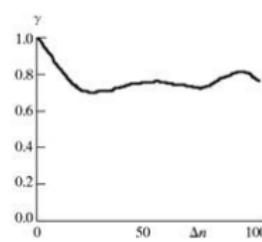
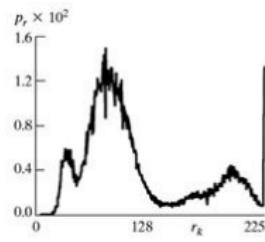
Избыточность в представлении изображений

- **Кодовая избыточность** возникнет из-за использования кодов, которые не минимизируют среднюю длину кодового слова (равномерный код вместо Хаффмана и т.д.).
- **Межпиксельная избыточность** связана с тем, что пиксель или группы пикселей похожи друг на друга.
 - ▶ Статистическая зависимость цветовых компонент (R, G и B).
 - ▶ Локальная схожесть соседних пикселей (local similarity).
 - ▶ Схожесть удалённых групп пикселей (non-local similarity).
- **Психовизуальная избыточность** зрительная система человека имеет разную чувствительность к визуальной информации (разное восприятие вертикальных, горизонтальных и диагональных линий, приоритет одних участков (лиц) изображения над другими).
- **Машинная избыточность** – аналогично, если предполагается, что изображение передаётся для машинной обработки, а не для человека.

Основные принципы сжатия изображений

Локальная и не локальная схожесть соседних пикселей

$$\gamma(\Delta n) = \frac{A(\Delta n)}{A(0)}, \quad A(\Delta n) = \frac{1}{N - \Delta n} \sum_{x=0}^{N-1-\Delta.n} f(x)f(x + \Delta n).$$



a)

b)

c)

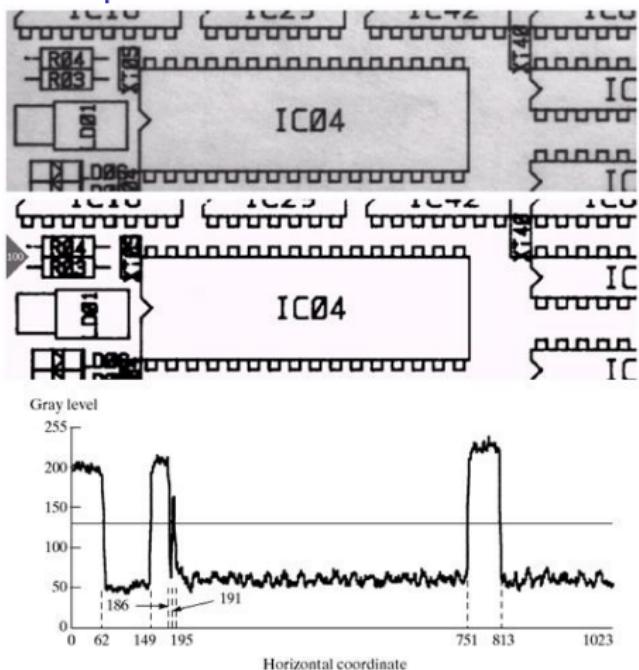
Рис.: а) Изображение, б) Гистограмма с) Автокорреляция для одной строки¹

¹R.Gonzalez, R.Woods, Digital Image Processing, 2001.

Основные принципы сжатия изображений

Сжатие бинарных изображений

- 1 1024×343 монохромное изображение преобразуется в бинарное²;
- 2 Каждая линия представляется парами (g_i, w_i) , где g_i – значение и w_i – количество повторений (длина серии).
- 3 Для линии 100 достаточно 88 для представления строки из 1024 значений.
- 4 Каждая серия представляется 11 битами, всего кодируется 1266 серий.
- 5 $C_R = \frac{1024 \cdot 343 \cdot 1}{1266 \cdot 11} = 2.63$



Line 100: (1, 63) (0, 87) (1, 37) (0, 5) (1, 4) (0, 556) (1, 62) (0, 210)

²R.Gonzalez, R.Woods, Digital Image Processing, 2001.

Основные принципы сжатия изображений

Thatcher effect



Основные принципы сжатия изображений

Thatcher effect



Основные принципы сжатия изображений

Region-of-interest coding³



a)



b)

Рис.: а) JPEG2000 б) JPEG2000+ROI

³A. Nguyen et. all, Gaze Tracking for Region of Interest Coding in JPEG2000, 2005. ↗ ↘

Основные принципы сжатия изображений

Оценка качества изображений

- Используются объективные и субъективные оценки качества изображений.
- Объективные метрики (MSE, PSNR)
 - ▶ Среднеквадратическая ошибка (MSE).

$$MSE = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - \hat{f}(x, y)]^2.$$

- ▶ Пиковое отношение сигнала к шуму (Peak signal-to-noise ratio, PSNR).

$$PSNR = 10 \lg \frac{|f_{max}(x, y)|^2}{MSE},$$

где $f_{max}(x, y)$ – максимально возможное значение яркости. Для 8-битного изображения $f_{max}(x, y) = 255$.

Основные принципы сжатия изображений

Субъективная оценка качества изображений

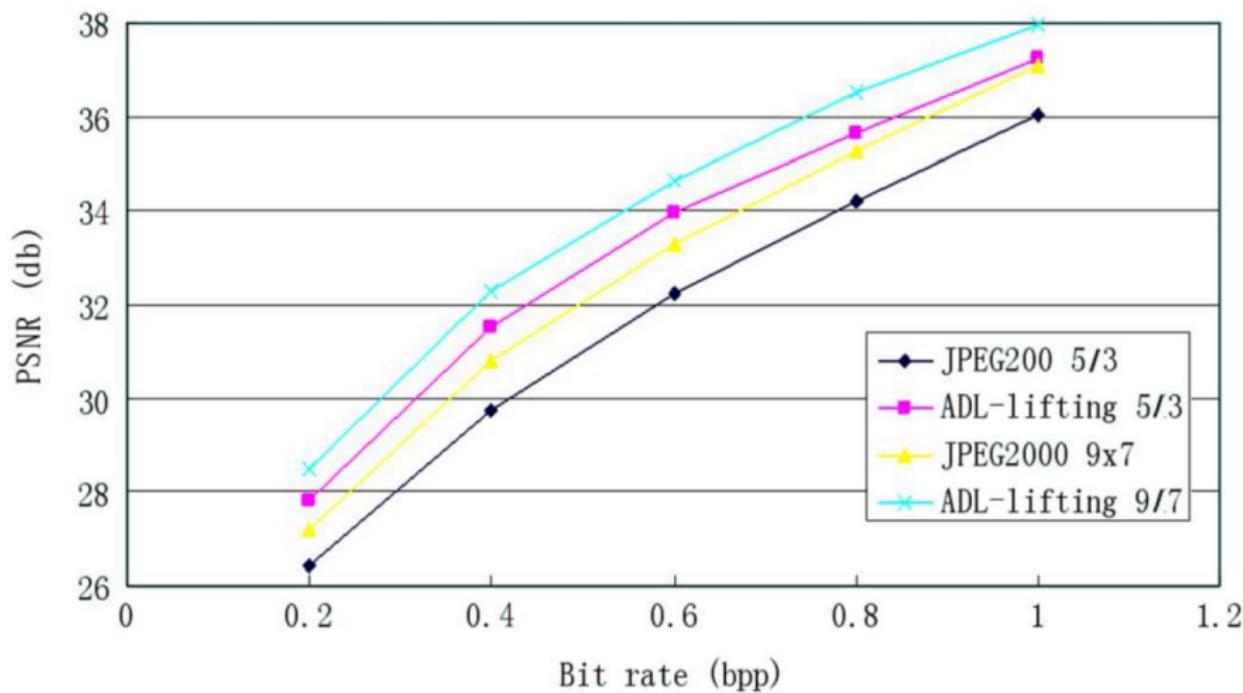
- Визуальное сравнение двух изображений:
 $\{-3, -2, -1, 0, 1, 2, 3\} = \{\text{much worse, worse, slightly worse, the same, slightly better, better, much better}\}$;
- Оценка одного изображения по шкале:⁴

Five-grade scale	
Quality	Impairment
5 Excellent	5 Imperceptible
4 Good	4 Perceptible, but not annoying
3 Fair	3 Slightly annoying
2 Poor	2 Annoying
1 Bad	1 Very annoying

⁴Methodology for the Subjective Assessment of the Quality of Television Pictures,
ITU-R Rec.BT.500-13, 2012

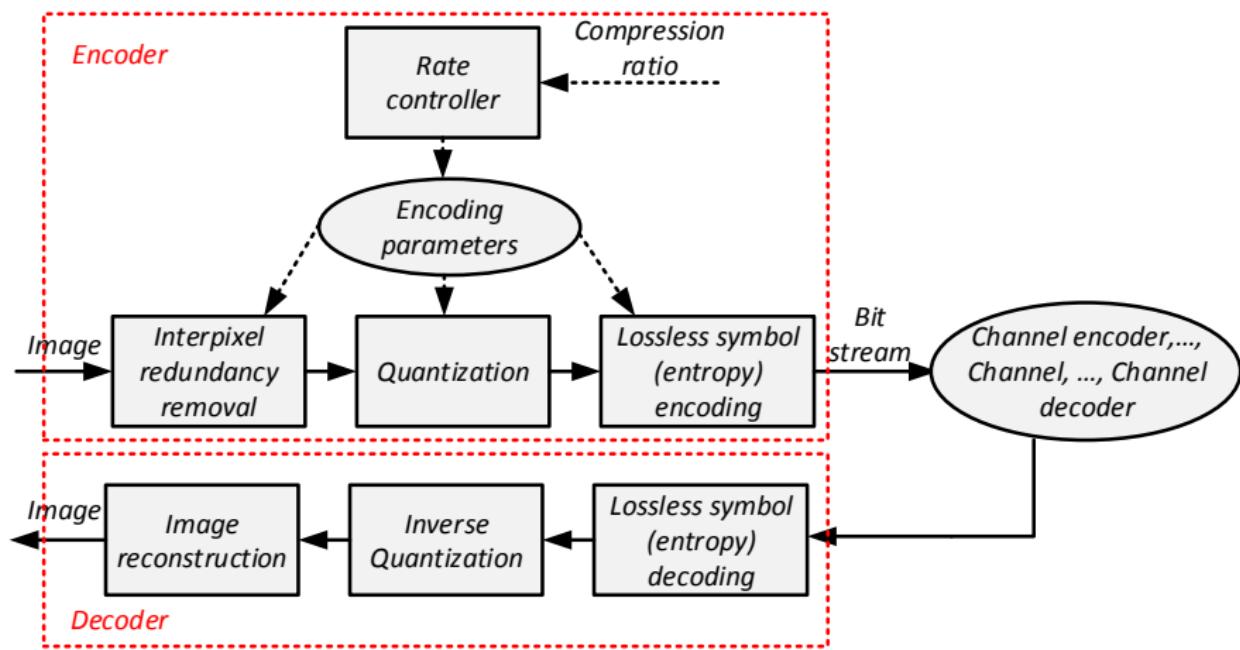
Основные принципы сжатия изображений

Функция скорость-искажение



Основные принципы сжатия изображений

Общая схема сжатия изображений



Основные принципы сжатия изображений

Общая схема сжатия изображений

- Устранение избыточности, связанной со схожестью пикселей:
 - ▶ Преобразование цветового пространства;
 - ▶ Кодирование с предсказанием;
 - ▶ Кодирование с преобразованием.
- Квантование (при сжатии с потерями);
- Адаптивное кодирование полученных данных:
 - ▶ Кодирование длин серий;
 - ▶ Побуквенное кодирование (Хаффман);
 - ▶ Контекстное адаптивное арифметическое кодирование.

Преобразование цветового пространства

Формат YCbCr

- Наиболее часто используется преобразование из формата RGB24 в формат YCbCr 4:2:0:
- Прямое преобразование:

$$\begin{cases} Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B, \\ Cb = (B - Y) \cdot 0.5643 + 128, \\ Cr = (R - Y) \cdot 0.7132 + 128. \end{cases}$$

- Обратное преобразование:

$$\begin{cases} G = Y - 0.714 \cdot (Cr - 128) - 0.334 \cdot (Cb - 128), \\ R = Y + 1.402 \cdot (Cr - 128), \\ B = Y + 1.772 \cdot (Cb - 128). \end{cases}$$

Преобразование цветового пространства

Пример



Original
Image



R



G



B



Original
image



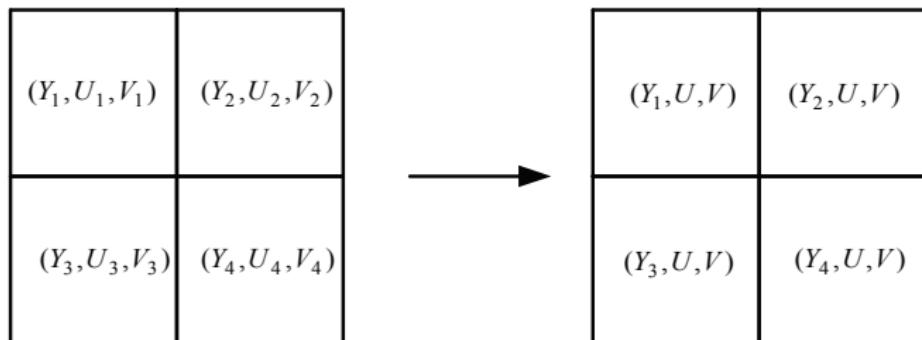
Luma (Y)



Chroma (Cb, Cr)

Преобразование цветового пространства

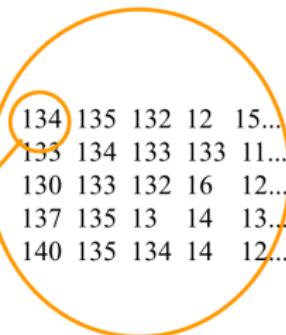
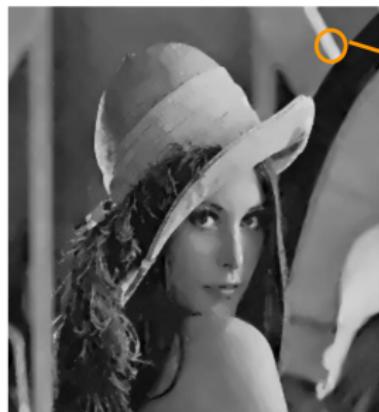
Децимация цветоразностных компонент Сb и Cr



$$U = \left\lfloor \frac{U_1 + U_2 + U_3 + U_4 + 2}{4} \right\rfloor, V = \left\lfloor \frac{V_1 + V_2 + V_3 + V_4 + 2}{4} \right\rfloor.$$

Простой кодек изображений

Монохромное изображение.



$134 = 10000110$ (1 byte or 8 bits per pixel)

- Каждый пиксель $x_i \in \{0, 1, \dots, 255\}$.

Простой кодек изображений

Квантование и энтропийное кодирование.

- ① На входе монохромное изображение $X = \{x_1, x_2, \dots\}$,
 $x_i \in \{0, 1, \dots, 255\}$.

② Кодер

- ① Квантование:
$$z_i = \left\lfloor \frac{x_i + \Delta/2}{\Delta} \right\rfloor.$$

- ② Кодирование $\{z_1, z_2, \dots\}$ (например, кодом Хаффмана).

- ③ Битрейт R можно оценить как: $R \approx H(Z) = -\sum_j p_j \log_2(p_j)$, где p_j

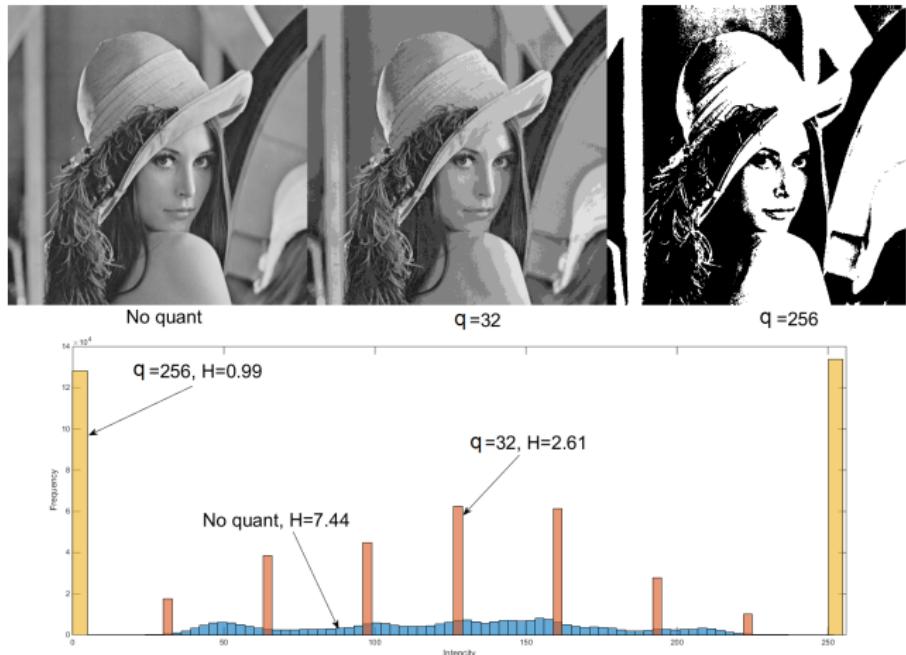
это вероятность $z_i = j$.

④ Декодер

- ① Декодирование $\{z_1, z_2, \dots\}$.
- ② Восстановление $\hat{x}_i = \Delta \cdot z_i$.

Простой кодек изображений

Квантование и энтропийное кодирование



- Квантование – необратимая операция, поэтому $H(Q(X)) < H(X)$.

Кодирование с предсказанием

Общая идея



Рис.: Исходное изображение

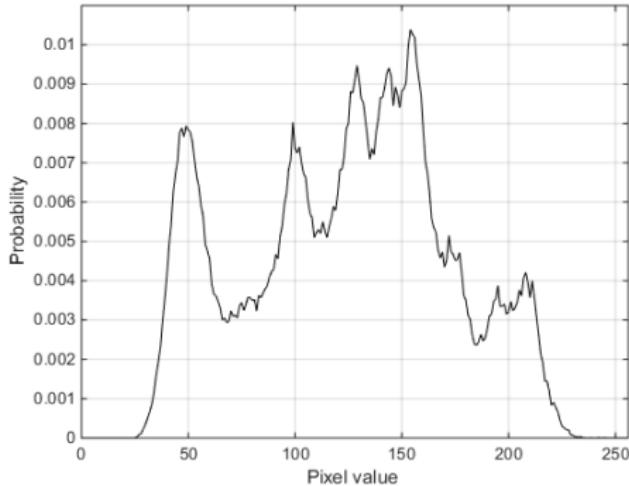


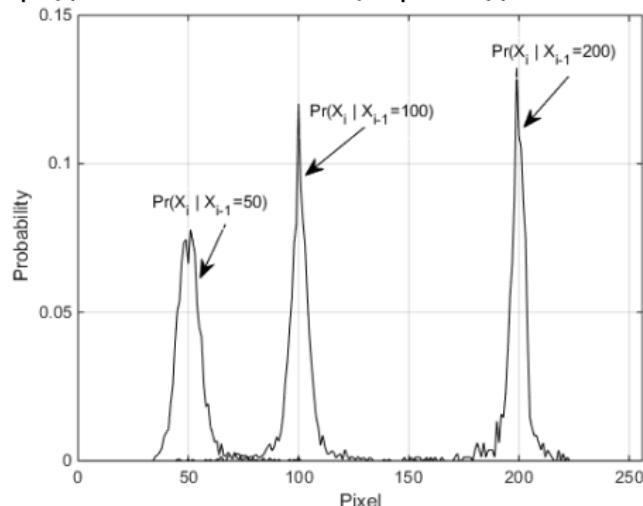
Рис.: Гистограмма X , $H(X) = 7.44$

- Код Хаффмана для данного распределения обеспечит не более $H(X) = 7.44$ бит на пиксель (без сжатия – 8 бит).

Кодирование с предсказанием

Общая идея

- Построим распределение пикселя x_i при заданном пикселе x_{i-1} .



- $H(X_i|50) = 4.56, H(X_i|100) = 4.57, H(X_i|200) = 4.27.$
- $R(X_i|X_{i-1}) \approx H(X_i|X_{i-1}) = \sum_{s=0}^{255} p(s)H(X_i|s) = 4.54 < H(X_i) = 7.44.$
- Код Хаффмана потребует 256 кодовых таблиц!

Кодирование с предсказанием

Общая идея

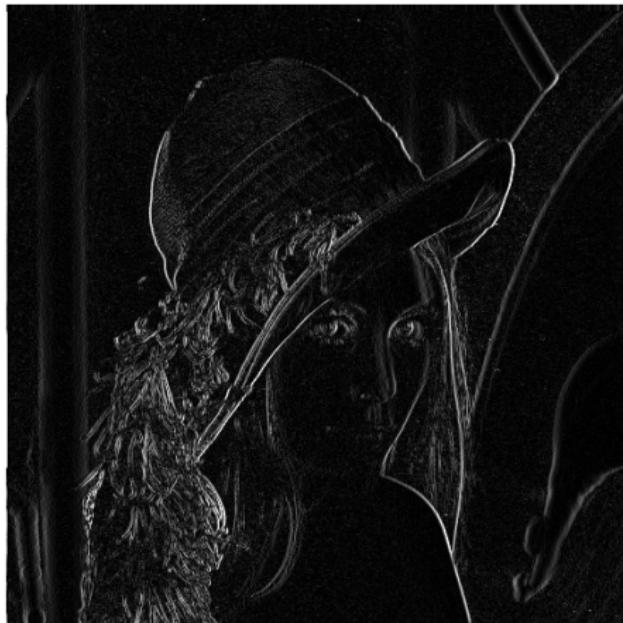


Рис.: Ошибка предсказания

$$e_i = x_i - \hat{x}_{i-1}$$

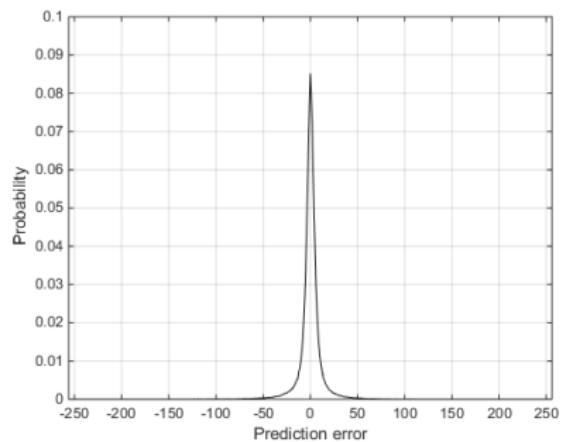


Рис.: Гистограмма ошибки предсказания, $H(E) = 5.07$

Кодирование с предсказанием

"Неправильный" алгоритм кодирования

- Кодер

- ➊ Вычисление ошибки предсказания: $e_i = x_i - x_{i-1}$
- ➋ Квантование: $z_i = \left\lfloor \frac{e_i + \Delta/2}{\Delta} \right\rfloor$
- ➌ Кодирование z_i .

- Декодер

- ➊ Декодирование z_i .
- ➋ Деквантование: $\hat{e}_i = \Delta \cdot z_i$.
- ➌ Восстановление: $\hat{x}_i = \hat{x}_{i-1} + \hat{e}_i$.

Накопление ошибки в декодере:

- ➊ Пусть $\hat{x}_0 = x_0$
- ➋ $e_1 = x_1 - x_0, (x_0 = x_1 - e_1)$
- ➌ $\hat{e}_1 = \Delta \cdot z_1 = e_1 + \delta_1$
- ➍ $\hat{x}_1 = x_0 + \hat{e}_1 = x_0 + e_1 + \delta_1 = x_1 + \delta_1$
- ➎ $e_2 = x_2 - x_1, (x_1 = x_2 - e_2)$
- ➏ $\hat{e}_2 = \Delta \cdot z_2 = e_2 + \delta_2$
- ➐ $\hat{x}_2 = \hat{x}_1 + \hat{e}_2 = x_1 + \delta_1 + e_2 + \delta_2 = x_2 + \delta_1 + \delta_2$
- ➑ $\hat{x}_n = x_n + \sum_{k=1}^n \delta_k.$



Кодирование с предсказанием

Кодирование с предсказанием и квантованием

- Кодер

- 1 Вычисление ошибки предсказания: $e_i = x_i - \hat{x}_{i-1}$.
- 2 Квантование: $z_i = \left\lfloor \frac{e_i + \Delta/2}{\Delta} \right\rfloor$.
- 3 Кодирование z_i .
- 4 Восстановление:
 $\hat{x}_i = \hat{x}_{i-1} + \hat{e}_i = \hat{x}_{i-1} + z_i \cdot \Delta$.

- Декодер

- 1 Декодирование z_i .
- 2 Деквантование: $\hat{e}_i = \Delta \cdot z_i$.
- 3 Восстановление: $\hat{x}_i = \hat{x}_{i-1} + \hat{e}_i$.

Ошибка в декодере:

- 1 $e_1 = x_1 - x_0$, ($x_0 = x_1 - e_1$)
- 2 $\hat{e}_1 = \Delta \cdot z_1 = e_1 + \delta_1$
- 3 $\hat{x}_1 = x_0 + \hat{e}_1 = x_0 + e_1 + \delta_1 = x_1 + \delta_1$
- 4 $e_2 = x_2 - \hat{x}_1$, ($\hat{x}_1 = x_2 - e_2$)
- 5 $\hat{e}_2 = \Delta \cdot z_2 = e_2 + \delta_2$
- 6 $\hat{x}_2 = \hat{x}_1 + \hat{e}_2 = \hat{x}_1 + e_2 + \delta_2 = x_2 + \delta_2$.
- 7 $\hat{x}_n = x_n + \delta_n$.

Кодирование с предсказанием

Сравнение кодирования с предсказанием и без предсказания



No quant, H=7.44



$q = 32$, H=2.61



$q = 256$, H=0.99



Pred+ No quant, H=5.01



Pred+ $q=32$, H=1.01



Pred+ $q = 256$, H=0.36



Кодирование с предсказанием

Сравнение кодирования с предсказанием и без предсказания



No quant., H=7.44



$Q = 32$, $H = 2.61$



$Q = 256$, $H = 0.99$



Pred+ No quant., $H = 5.01$



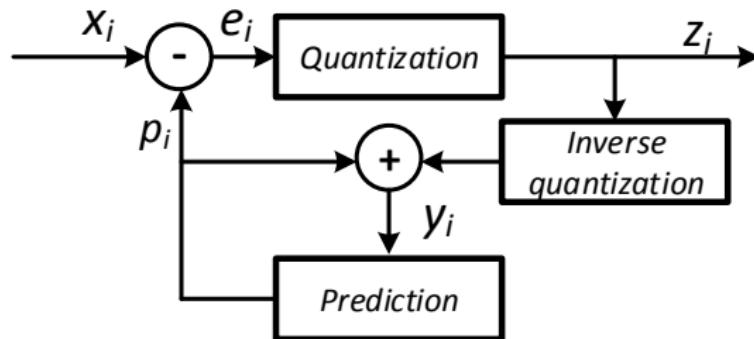
Pred+ $Q = 6$, $H = 2.60$



Pred+ $Q = 32$, $H = 1.01$

Кодирование с предсказанием

Общая схема



- ① Вычисление ошибки предсказания: $e_i = x_i - p_i$.
- ② Квантование: $z_i = \left\lfloor \frac{e_i + q/2}{q} \right\rfloor$.
- ③ Кодирование z_i .
- ④ Деквантование: $y_i = \hat{e}_i + p_i = z_i \cdot q + p_i$.
- ⑤ Вычисление предсказателя: $p_{i+1} = f(y_0, y_1, \dots, y_i)$.
- ⑥ Простейший случай: $p_{i+1} = y_i$.

Кодирование с предсказанием

Стандарт JPEG-LS⁵

c	b	d
a	x	

JPEG-LS работает в двух режимах: *regular mode* и *run-length mode*.

- Вычисление градиента:

$$\begin{cases} d_1 = d - b, \\ d_2 = b - c, \\ d_3 = c - a. \end{cases}$$

- Если $d_1 = d_2 = d_3 = 0$, то кодер переходит в режим кодирования длин серий до тех пор, пока не произойдёт $a \neq x$, либо не будет достигнут конец строки. Длина серии передаётся монотонным кодом (Rice-Golomb code) и кодер возвращается в regular mode.

⁵ISO/IEC 14495-1, ITU Recommendation T.87, "Information technology - Lossless and near-lossless compression of continuous-tone still images," 1999.

Кодирование с предсказанием

Стандарт JPEG-LS. Regular mode

c	b	d
a	x	

- Пиксели обрабатываются в растровом порядке;
- Вычисление предсказателя:

$$p_x \leftarrow \begin{cases} \min(a, b) & \text{if } c \geq \max(a, b), \\ \max(a, b) & \text{if } c \leq \min(a, b) \\ a + b - c & \text{otherwise.} \end{cases}$$

- Коррекция предсказателя:

$$p_x \leftarrow p_x + \Delta p(d1, d2, d3).$$

- Ошибка предсказания $e_x \leftarrow x - p_x$ кодируется монотонным кодом (Rice-Golomb code).

Кодирование с предсказанием

Общая схема стандарта JPEG-LS⁶

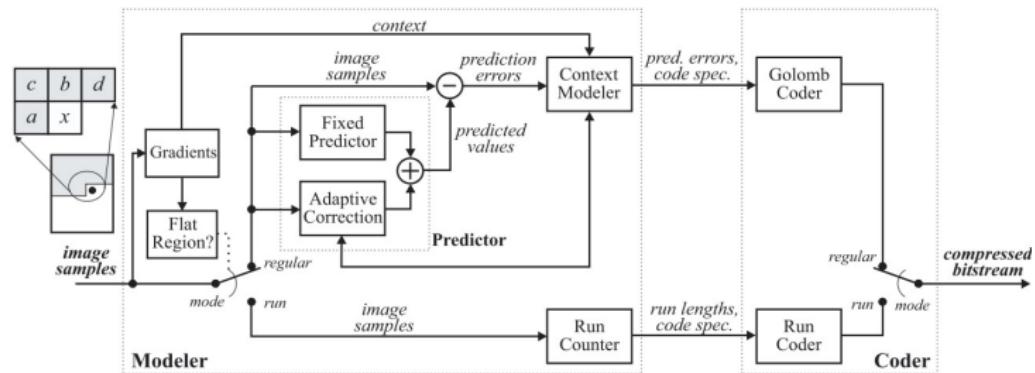


Таблица: Биты на пиксель для "Lena"

CALIC	JPEG-LS	JPEG2000 lossless	WinRAR
4.121	4.237	4.330	5.135

⁶M. Weinberger, G. Seroussi, G. Sapiro, The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS.

Кодирование с преобразованием

Основная идея

- Делим изображение на непересекающиеся блоки размером $n \times n$;
- Выполняем преобразование $n \times n$ для каждого блока:
 - ▶ Преобразование должно декорелировать значения в блоке или представить информацию о блоке в наименьшем количестве коэффициентов преобразования.
 - ▶ Преобразование должно быть ортонормальным (искажение в области коэффициентов преобразования должно быть равно искажению в области сигнала).
 - ▶ Преобразование должно иметь небольшую вычислительную сложность (например, быть сепарабельным).
- Квантование устраняет наименее информативные коэффициенты преобразования.
- Энтропийное кодирование применяется к квантованным коэффициентам преобразования.

Кодирование с преобразованием

Выбор преобразования

- Прямое преобразование

$$T(u, v) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y)g(x, y, u, v),$$

$$u, v = 0, 1, 2, \dots, N - 1.$$

- Обратное преобразование

$$f(x, y) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} T(x, y)h(x, y, u, v).$$

- Сепарабельное преобразование:

$$g(x, y, u, v) = g_1(x, u) \cdot g_2(y, v).$$

- Симметричное преобразование:

$$g(x, y, u, v) = g_1(x, y) \cdot g_1(u, v).$$

Кодирование с преобразованием

Выбор преобразования

Возможные кандидаты:

- Преобразование Карунена-Лоева (Karhunen-Loeve Transform, KLT);
 - ▶ Гарантирует, что коэффициенты не коррелированы.
 - ▶ Оптимальный базис зависит от входных данных, т.е., его нужно передавать декодеру.
 - ▶ Высокая вычислительная сложность;
- Дискретное преобразование Фурье (Discrete Fourier Transform, DFT);
 - ▶ Имеет избыточные (мнимые) коэффициенты;
- Дискретное косинусное преобразование (Discrete Cosine Transform, DCT);
- Дискретное преобразование Уолша-Адамара (Walsh-Hadamard transform, WHT);

Кодирование с преобразованием

Преобразование Уолша-Адамара

$$g(x, y, u, v) = h(x, y, u, v) = \frac{1}{N}(-1)^{(b_i(x)p_i(u) + b_i(y)p_i(v))}, N = 2^m,$$

where $b_i(x)$ – значение бита в x на позиции i .

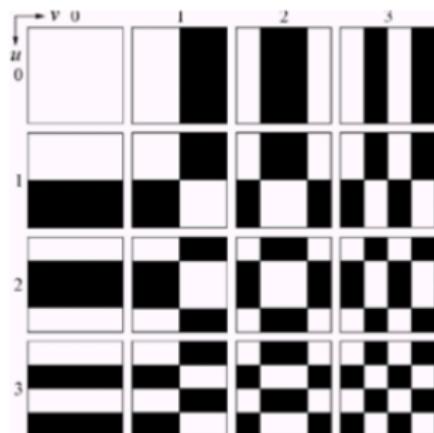
$$p_0(u) = b_{n-1}(u),$$

$$p_1(u) = b_{n-1}(u) + b_{n-2}(u),$$

$$p_2(u) = b_{n-2}(u) + b_{n-3}(u),$$

...

$$p_{n-1}(u) = b_1(u) + b_0(u).$$

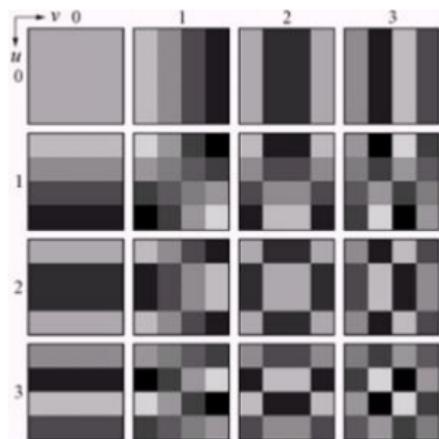


Кодирование с преобразованием

Дискретное косинусное преобразование

$$g(x, y, u, v) = h(x, y, u, v) = a(u)a(v) \cos\left(\frac{(2x+1)u\pi}{2N}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right)$$

$$a(u) = \begin{cases} \frac{1}{\sqrt{N}}, & u = 0 \\ \frac{2}{\sqrt{N}}, & u \neq 0. \end{cases}$$



Кодирование с преобразованием

Дискретное преобразование Фурье

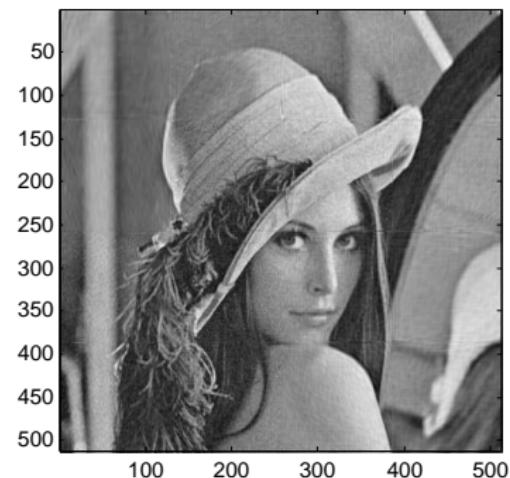
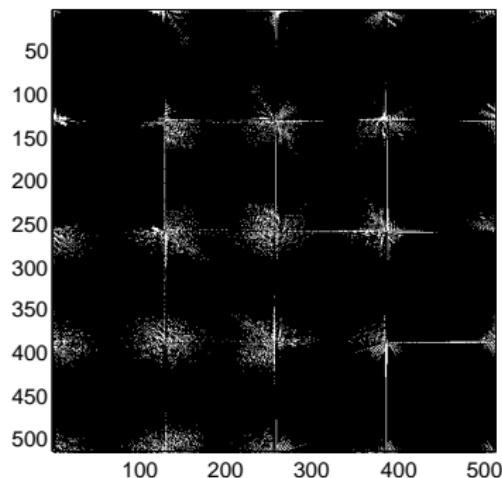


Рис.: FFT 128×128 после удаление 90% коэффициентов с наименьшей амплитудой, Y-PSNR=31.23 дБ

Кодирование с преобразованием

Дискретное преобразование Уолша-Адамара

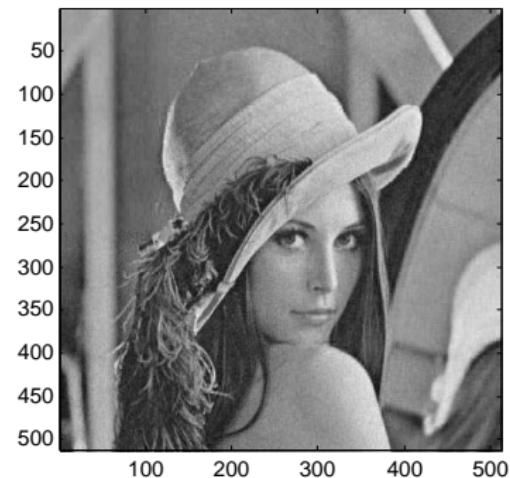
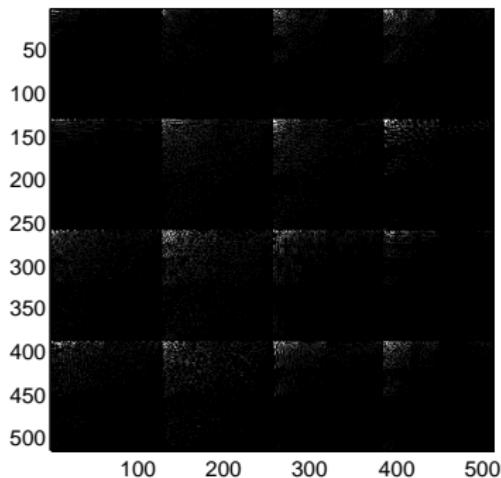


Рис.: WHT 128×128 после удаление 90% коэффициентов с наименьшей амплитудой, Y-PSNR=30.40 дБ

Кодирование с преобразованием

Дискретное косинусное преобразование

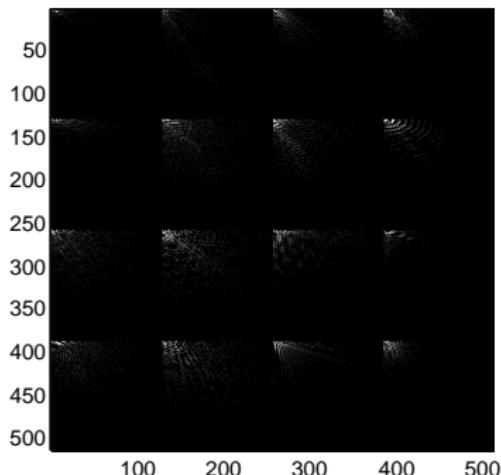


Рис.: DCT 128×128 после удаление 90% коэффициентов с наименьшей амплитудой, Y-PSNR=34.15 дБ

Кодирование с преобразованием

Выбор преобразования

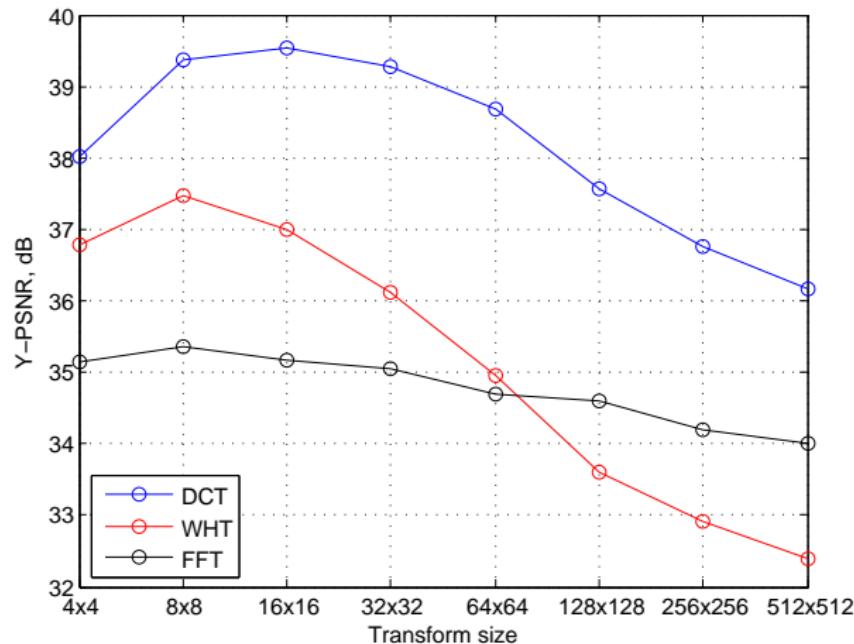


Рис.: Y-PSNR после удаление 80% коэффициентов с наименьшей амплитудой

Стандарт JPEG

Основные этапы ⁷

- ① Преобразование цветового пространства из RGB24 в YCbCr 4:2:0;
- ② Разбиение яркостной и цветоразностной компоненты на блоки 8×8 ;
- ③ Применение 2-D DCT для каждого блока;
- ④ Квантование DCT коэффициентов;
- ⑤ DC коэффициент из текущего блока предсказывается при помощи DC коэффициента предыдущего блока и кодирование разности кодом Левенштейна с кодом Хаффмана в первой части кодового слова.
- ⑥ Сканирование AC коэффициентов в 'зигзагообразном' порядке.
- ⑦ Одномерный вектор AC кодируется длинами серий и кодом Хаффмана.

⁷Digital compression and coding of continuous-tone still images, ITU-T and ISO/IEC JTC1, 1992.

Стандарт JPEG

Пример

Исходный блок 8×8 :

$$X = \begin{pmatrix} 168 & 161 & 161 & 150 & 154 & 168 & 164 & 154 \\ 171 & 154 & 161 & 150 & 157 & 171 & 150 & 164 \\ 171 & 168 & 147 & 164 & 164 & 161 & 143 & 154 \\ 164 & 171 & 154 & 161 & 157 & 157 & 147 & 132 \\ 161 & 161 & 157 & 154 & 143 & 161 & 154 & 132 \\ 164 & 161 & 161 & 154 & 150 & 157 & 154 & 140 \\ 161 & 168 & 157 & 154 & 161 & 140 & 140 & 132 \\ 154 & 161 & 157 & 150 & 140 & 132 & 136 & 128 \end{pmatrix}$$

Стандарт JPEG

Пример

Блок после вычитания 128 и выполнения 2-D DCT:

$$Y = \begin{pmatrix} 214 & 49 & -3 & 20 & -10 & -1 & 1 & -6 \\ 34 & -25 & 11 & 13 & 5 & -3 & 15 & -6 \\ -6 & -4 & 8 & -9 & 3 & -3 & 5 & 10 \\ 8 & -10 & 4 & 4 & -15 & 10 & 6 & 6 \\ -12 & 5 & -1 & -2 & -15 & 9 & -5 & -1 \\ 5 & 9 & -8 & 3 & 4 & -7 & -14 & 2 \\ 2 & -2 & 3 & -1 & 1 & 3 & -3 & -4 \\ -1 & 1 & 0 & 2 & 3 & -2 & -4 & -2 \end{pmatrix}$$

Стандарт JPEG

Пример

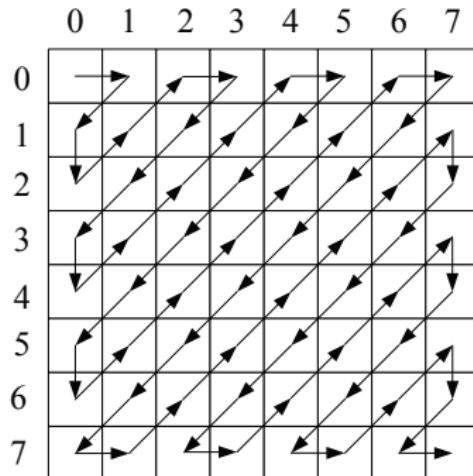
Блок после скалярного квантования:

$$Z = \begin{pmatrix} 13 & 4 & 0 & 1 & 0 & 0 & 0 & 0 \\ 3 & -2 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Стандарт JPEG

Пример

Коэффициенты сканируются в "зигзагообразном" порядке:



- После сканирования

$$Z = \{13, 4, 3, 0, -2, 0, 1, 1, 0, 1, -1, -1, 1, 1, 0, \dots, 0\};$$

Стандарт JPEG

Пример

- $Z = \{13, 4, 3, 0, -2, 0, 1, 1, 0, 1, -1, -1, 1, 1, 0, \dots, 0\}$;
- Первый коэффициент 13 (DC coefficient) предсказывается по DC коэффициенту из предыдущего (слева) блока, затем амплитуда ошибки предсказания кодируется монотонным кодом, в котором первая часть кодируется кодом Хаффмана. Дополнительный бит используется для передачи знака для ненулевых значений.

$n \rightarrow \underbrace{\text{huff(DC bits)}}_{\text{DC bit size}} \quad \underbrace{\text{bin(DC)}}_{\text{DC}} \quad \underbrace{\text{one bit}}_{\text{sign}}$.

Стандарт JPEG

Пример

- $Z = \{13, 4, 3, 0, -2, 0, 1, 1, 0, 1, -1, -1, 1, 1, 0, \dots, 0\}$;
- Оставшиеся коэффициенты (AC coefficients) представляются парами $[run, level]$, где run – число нулей перед ненулевым коэффициентом, $level$ – значение ненулевого коэффициента. В нашем случае:
 $[0, 4], [0, 3], [1, -2], [1, 1], [0, 1], [1, 1], [0, -1], [0, -1], [0, 1], [0, 1]$.
-

$[run, level] \rightarrow$ Huffman bin(|level|) one bit.
 [run, level bit size] level sign

- Если пара $[run, level]$ не присутствует в кодовой таблице Хаффмана, то передаётся ESC-символ, после чего run и $level$ передаются равномерным кодом.
- End-of-block (EOB) символ передаётся кодом Хаффмана, если в “зиг-заге” далее следуют только нули.

Стандарт JPEG

Пример

Блок после декодирования кодом Хаффмана, обратного прохода по "зиг-загу", деквантования, обратного преобразования и прибавления 128:

$$\hat{X} = \begin{pmatrix} 171 & 160 & 149 & 149 & 158 & 166 & 166 & 162 \\ 174 & 164 & 155 & 154 & 160 & 164 & 161 & 156 \\ 171 & 164 & 157 & 156 & 158 & 158 & 151 & 145 \\ 161 & 157 & 154 & 154 & 155 & 151 & 144 & 137 \\ 156 & 155 & 155 & 156 & 156 & 152 & 145 & 140 \\ 159 & 160 & 160 & 160 & 157 & 153 & 148 & 145 \\ 161 & 161 & 160 & 156 & 150 & 144 & 141 & 139 \\ 159 & 158 & 155 & 148 & 139 & 132 & 129 & 128 \end{pmatrix}$$

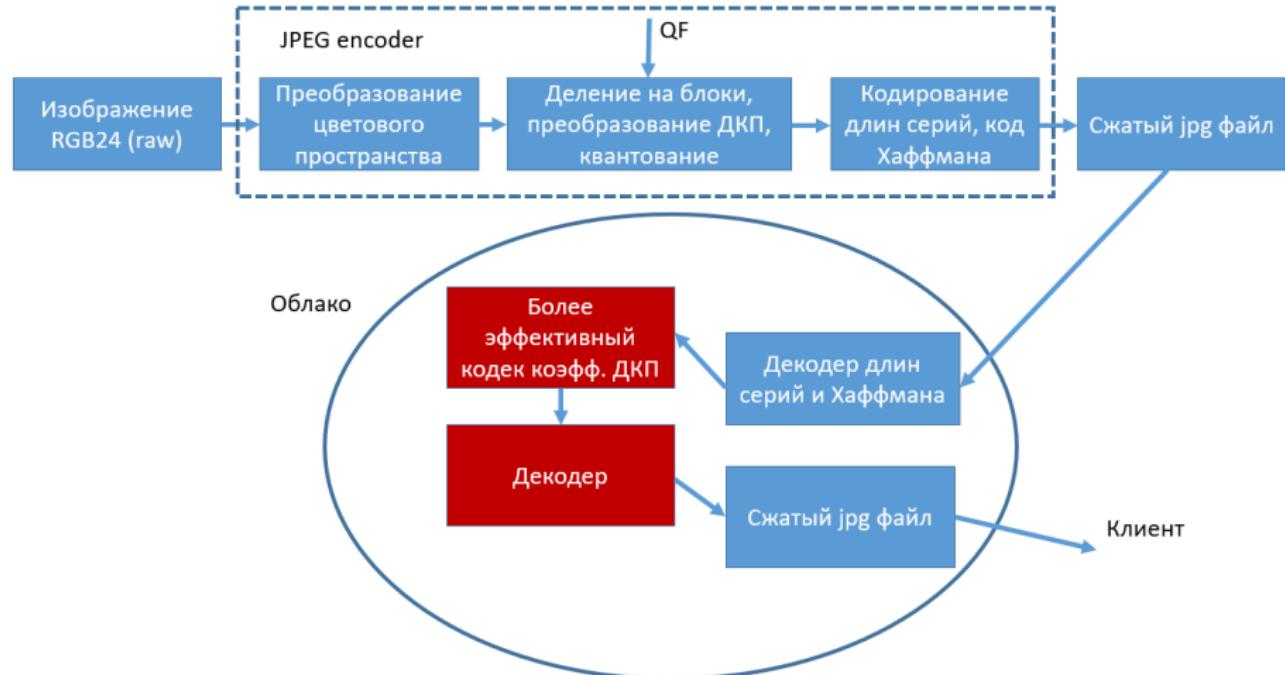
Стандарт JPEG

Блоковые артефакты

При больших значениях Δ возникают хорошо видимые границы блоков:



Лабораторная работа №2. Транскодер для формата jpg



Лабораторная работа №2. Транскодер для формата jpg

Тестовые изображения и размеры файлов формата JPEG

Изображение	Размер	JPEG, QF=80	JPEG, QF=30
airplane	512 × 512	44079	19207
arctichare	594 × 400	26569	12080
baboon	512 × 512	88465	36113
cat	490 × 733	81974	35902
fruits	512 × 512	45303	17646
frymire	1118 × 1105	440857	200860
girl	768 × 512	59979	24310
lena	512 × 512	43872	17578
monarch	768 × 512	64055	28501
peppers	512 × 512	47929	18746
pool	510 × 383	14492	7467
sails	768 × 512	105830	45512
serrano	629 × 794	138167	58288
tulips	768 × 512	85764	37972
watch	1024 × 768	101074	46741

Лабораторная работа №2. Транскодер для формата jpg

Требования к программе и отчету

Требования к программе:

- ① Написать кодер и декодер, которые работают как отдельные программы.
- ② В качестве параметра на вход кодера подаётся имя транскодируемого файла в формате jpg. На выходе программа выдаёт файл со сжатыми данными.
- ③ На вход декодера подаётся файл со сжатыми данными. На выходе декодер выдаёт файл, который идентичен исходному файлу в формате jpg (бит в бит).
- ④ Алгоритмы, относящиеся непосредственно к кодированию и декодированию должны быть реализованы без использования сторонних библиотек.
- ⑤ При улучшении сжатия на 2% и более для QF=30, начисляется 10 баллов. При улучшении сжатия на 2% и более для QF=80, начисляется 10 баллов.

Требования к отчету:

- ① Отчет в pdf формате, который включает в себя:
 - ▶ ФИО студента, номер группы и задания.
 - ▶ Описание алгоритма кодирования и декодирования.
 - ▶ Для каждого из 15-ти изображений указать размер сжатого файла в байтах.
Данные представить в виде таблицы.
 - ▶ Суммарное значение сжатого размера всех файлов в байтах.
- ② Исходный код кодера и декодера.
- ③ Исполняемые программы кодера и декодера, готовыми к запуску на Windows 10.

