

# Математические методы обработки изображений

Подрезов Дмитрий Евгеньевич, аспирант,  
Научный руководитель  
д.ф.-м.н., проф. Пилиди Владимир Ставрович

# Содержание доклада

- Основная цель хранения информации
- Алгоритмы сжатия
- Представление изображений
- Сжатие изображений
- Обработка изображений

# Основная цель хранения информации: проблема

- Основная задача хранения информации - экономия места
  - $1024 \times 768 \times 3 = 2304 \text{ Кб} = 2 \text{ Мб}^*$ 
    - \*(16777216 цветов)
- Для трехминутного видео
  - 5400 фреймов = почти 12 Гб
- Вывод: Необходимо избавляться от избыточности

# Основная цель хранения информации: отличие изображений от текста

- Три отличия изображений от текста:
- В тексте соседние символы слабо коррелируют, пиксели в изображении - сильно.
- В тексте вероятность появления символа, на изображении вероятность пикселей сильно зависит от характера изображения в данной области.
- Изображения редактируются редко, а просматриваются часто.

# Алгоритмы сжатия: принцип сжатия изображений

- Принцип сжатия изображений
- Если случайно выбрать пиксел изображения, то с большой вероятностью ближайшие к нему пиксели будут иметь тот же или близкий цвет.
- При этом переход цвета в одном направлении будет происходить по регулярному правилу.

# Алгоритмы сжатия: оценки

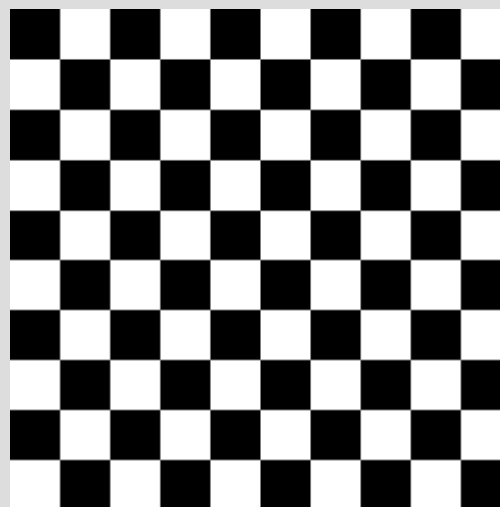
$$\text{Коэффициент сжатия} = \frac{\text{размер выходного файла}}{\text{размер входного файла}}$$

- bpb - bit per bit
- bpp - bit per pixel
- bpc - bit per character
- bitrate - битовая скорость
- bit budget - битовый бюджет

# Алгоритмы сжатия: коды Хаффмана

# Алгоритмы сжатия: run-length encoding

- Кодировются серии величин
- Худший вариант:



- бюджет - либо словарь, либо вероятности, либо  $\emptyset$



# Алгоритмы сжатия: декодирование

- Строится дерево аналогичным образом, как и при кодировании
- Производится проход по символам
- Адаптивное кодирование – таблица строится в процессе кодирования
- Проблема: отличить очередной блок данных от индекса в таблице
- Решение: escape-последовательности

# Алгоритмы сжатия: факсимильная связь

- Пел (pel) - picture element
- Чаще всего встречаются серии 2,3,4 черных пелов и 2-7 белых пелов
- 0 00110101 0000110111
- 1 000111 010
- 2 0111 11
- 2 1000 10
- 4 1011 011
- 5 1100 0011
- 6 1110 0010
- 7 1111 00011
- 8 1011 000101
- 9 10100 000100
- ...
- 1664 011000 0000001100100

# Алгоритмы сжатия: LZ77

- Метод скользящего окна - [Ziv 77]
- Зив, Лемпэл
- ...[sir sid eastman easily t][eases sea sick seals] when...
- (16,3,"e")
- ...sir [sid eastman easily tease][s sea sick seals whe]n...

# Алгоритмы сжатия: LZ77

- [][sir sid eastman ] (0,0,"s")
- [s][ir sid eastman e] (0,0,"i")
- [si][r sid eastman ea] (0,0,"r")
- [sir][ sid eastman eas] (0,0," ")
- [sir ][sid eastman easi] (4,2,"d")
- [sir sid][ eastman easily ] (4,1,"e")
- [sir sid e][astman easily te] (0,0,"a")
- ...Mr. [alf eastman easily grows alf][alfa in his ]garden...
- (3,4," ")
- ...Mr. [alf eastman easily grows alf][xxxxxxxxxxxx]garden...
- ...Mr. [alf eastman easily grows alf][axxxxxxxxxxxx]garden...
- ...Mr. [alf eastman easily grows alf][alxxxxxxxxxxx]garden...
- ...Mr. [alf eastman easily grows alf][alfxxxxxxxxxxx]garden...
- ...Mr. [alf eastman easily grows alf][alfaxxxxxxxxxx]garden...
- ...Mr. [alf eastman easily grows alf][alfa xxxxxxxx]garden...

# Алгоритмы сжатия: LZ78

- 1 "s" (0,"s")
- 2 "i" (0,"i")
- 3 "r" (0,"r")
- 4 " " (0," ")
- 5 "si" (1,"i")
- 6 "d" (0,"d")
- 7 " e" (4,"e")
- 8 "a" (0,"a")
- 9 "st" (1,"t")
- 10 "m" (0,"m")
- 11 "an" (8,"n")
- 12 " ea" (7,"a")
- 13 "sil" (5,"l")
- 14 "y" (0,"y")
- 15 " t" (4,"t")
- 16 "e" (0,"e")
- 17 "as" (8,"s")
- 18 "es" (16,"s")
- 19 " s" (4,"s")
- 20 "ea" (4,"a")
- 21 " si" (19,"i")
- 22 "c" (0,"c")
- 23 "k" (0,"k")
- 24 " se" (19,"e")
- 25 "al" (8,"l")
- 26 "s" (1,"")

# Алгоритмы сжатия: LZW

- [Welch 84]
- Словарь
- “a”, “b”, “c”, ... , “z”, “ea”, “si”, ...

# Представление изображений: избыточность

- Сколько информации?
- 2 байта



# Представление изображений: цветовые схемы

- CIE, 1931 – хроматические диаграммы
- RGB – red, green, blue  
[0..255]x[0..255]x[0..255]
  - Возможен случай, когда два соседних пикселя имеют различный оттенок, но при этом одинаковую яркость.
- Чтобы учитывать это, переведем представление RGB в YCbCr:
  - Y - яркость
  - Cb и Cr определяют оттенок.



# Представление изображений: цветовые схемы

- $Y = (77/256)R + (150/256)G + (29/256)B,$
- $Cb = -(44/256)R - (87/256)G + (131/256)B + 128,$
- $Cr = (131/256)R - (110/256)G - (21/256)B + 128.$

# Представление изображений: частные случаи

- Частные случаи:
  - монохроматическое изображение
  - полутоновое изображение
  - цветное изображение
  - изображение с непрерывным тоном
  - дискретно-тоновое изображение
  - анимации

# Сжатие изображений: ПОДХОДЫ

1. RLE
2. Контекстный
3. Раздельное кодирование RLE по слоям
  1. Проблема  $7=0111$ ,  $8=1000$ .
  2. Решение - коды Грея
4. Используем контекст  $A$ , предполагаем  $P = A$ ,  $\Delta = P - A$ , для  $\Delta$  используем префиксный код
  - $P = 0..m-1$ ,  $\Delta = -(m-1) .. m-1$ ,  $2m-1$  кодов
  - $\Delta$  распределены близко к распределению Лапласа
5. Wavelet
6. Производить преобразование 2, 3 или 4 с каждой компонентой
7. Разделение изображения на блоки и отдельные блоки кодировать по-разному
8. Фрактальные методы

# Сжатие изображений: матричные преобразования

- Декоррелирование

- $C = W * D$

$$W = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{pmatrix}$$

# Сжатие изображений: DCT

- Дискретное косинус-преобразование

$$w(f) = \cos(f\theta) \quad \mathbf{p} = \sum_i w_i \mathbf{v}_i$$

- (11, 22, 33, 44, 55, 66, 77, 88) →
- (140, -71, 0, -7, 0, -2, 0, 0)
- (140, -71, 0, -7, 0, -2, 0, 0) →
- (15, 20, 30, 43, 56, 69, 79, 84)

# Сжатие изображений: DCT

$$G_{ij} = \frac{1}{\sqrt{2n}} C_i C_j \sum_{x=0}^{n-1} \sum_{y=0}^{n-1} p_{xy} \cos \left( \frac{(2y+1)j\pi}{2n} \right) \cos \left( \frac{(2x+1)i\pi}{2n} \right)$$

# Сжатие изображений: другие виды преобразований

- Уолша-Адамара
- Хаара
- Кархунена-Лоэве

# Сжатие изображений: JPEG

- Joint Photographic Expert Group
- Лучше всего подходит для изображений с плавным переходом цвета
- Применим для любых видов изображений
- Для изображения  $512 \times 512$  требуется 12582912 умножений (и столько же сложений), но у [Feig 90] есть вариант, для которого необходимо 221184 умножения и 1916928 сложений



# Сжатие изображений: JPEG

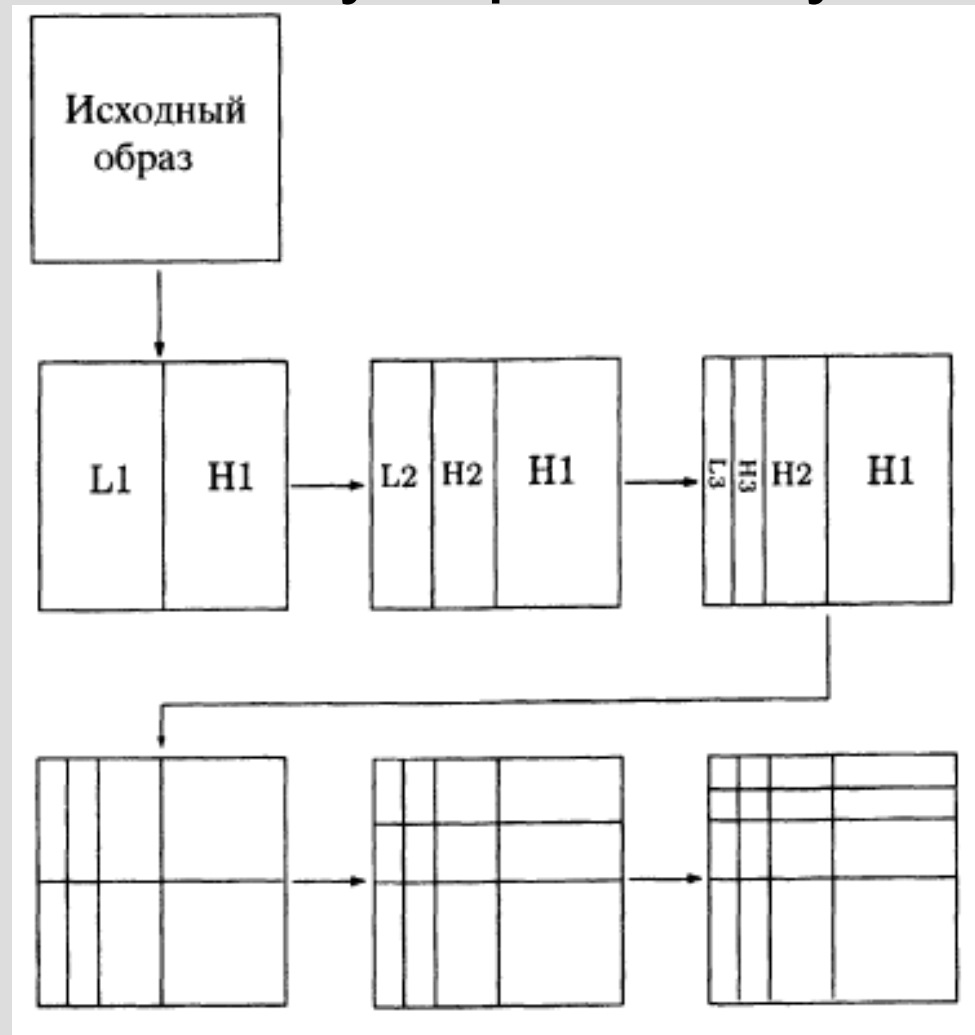
- Преобразование из RGB в YcrCb
- Разбиение на крупные пиксели (2h2v, 2h1v)
- Разбиение на блоки 8x8
- Дискретное косинус-преобразование
- Деление на коэффициент квантования (QC)
- Кодирование RLE
- Сохранение, добавление заголовков и т.д.

# Сжатие изображений: wavelet

- Полусуммы и полуразности:
  - (1,2,3,4,5,6,7,8)
  - (3/2,7/2,11/2,15/2,-1/2,-1/2,-1/2,-1/2)
  - (10/4,26/4,-4/4,-4/4,-1/2,-1/2,-1/2,-1/2)
  - (36/8,-16/8,-4/4,-4/4,-1/2,-1/2,-1/2,-1/2)
  - Нормализуем каждую компоненту (делим на корень из количества оставшихся средних)
  - (36/8√1, -16/8√1, -4/4√2, -4/4√2, -1/2√4, -1/2√4, -1/2√4, -1/2√4) =
  - (36/8, -2, -1/√2, -1/√2, -1/42, -1/42, -1/42, -1/4)
  - (5,-2,-1,-1,0,0,0,0)

# Сжатие изображений: wavelet

- Обобщение на двумерный случай:



# Литература

- **[Сэл 04]** Д. Сэломон, Сжатие данных, изображений и звука, Техносфера, 2004
- **[Feig 90]** E.N. Feig, E. Linzer, Discrete cosine transform algorithms for image data compression, Proceedings Electronic Imaging, 84-87, 1990
- **[Pratt ]** W. Pratt. Digital image processing, Wiley, 2007
- **[Welch 84]** T.A. Welch, A technique for high-performance data compression, IEEE 17(6):8-19, 1984
- **[LZ 77]** J. Ziv, A. Lempel, A universal algorithm for sequential data compression, IEEE 23(3):337-343, 1977
- **[LZ 78]** J. Ziv, A. Lempel, Compression of individual sequences via variable-rate coding, IEEE 24(5):530-536, 1978