

Форматы хранения изображений.

лекция 10

Старшие - представление информации в более экспрессивном виде, влекущее за собой увеличение объема данных (как правило). В основном алгоритмы старшие используют свойства графических данных:

- Избыточность - группы одинаковых символов.
- Предсказуемость - часто повторяющиеся одинаковые комбинации символов.
- Необязательность - данные, мало влияющие на человеческое восприятие.

Оценки методов хранения:

- Степень сжатия - отношение объема старого файла к объему нового файла.
- Точность восстановления - среднеквадратичная ошибка значений пикселей старого изображения от оригинала.
- Скорости компрессии и декомпрессии - среднее время сжатия и восстановления.
- Симметричность - отношение времени сжа-

мень до времени восстановления.

- Коэффициент точности: $K_{\text{отн}} = \frac{N}{N_{\text{отн}}}$

- Критерий среднеквадратичного отклонения:

$$\varepsilon = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

- Критерий максимального отклонения:

$$\varepsilon = \max_N |x_i - y_i|$$

Методы поиска

Без потерь информации

С частичной потерей информации

- Отсутствие стилиза. Формат BMP

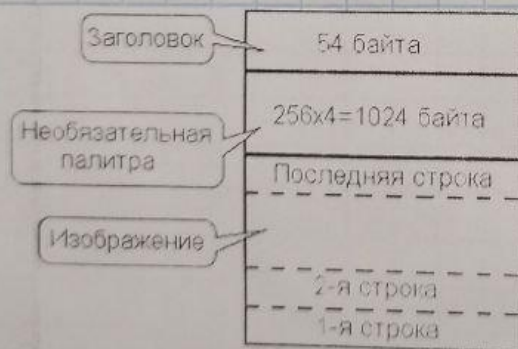


Рис. 4.1. Общая структура BMP-файла

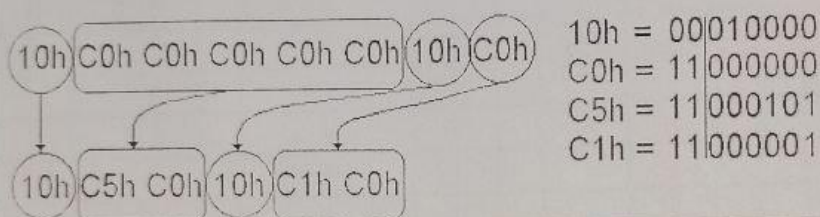
Формат заголовка BMP-файла

Наименование поля	Смещение	Длина	Описание
bfType	+00h	2	Сигнатура 'BM'
bfSize	+02h	4	Размер файла
bfReserved	+06h	4	Не используется
bfOffBits	+0Ah	4	Позиция битов изображения
biSize	+0Eh	4	40 - размер второй части заголовка
biWidth	+12h	4	Длина строки
biHeight	+16h	4	Количество строк
biPlanes	+1Ah	2	Количество слоев

biBitCount	+1Ch	2	Количество битов на пиксел
biCompression	+1Eh	4	0 – сжатия нет
biSizeImage	+22h	4	Размер картинки в битах
biXPelsPerMeter	+26h	4	Разрешение по X в пикселах на метр
biYPelsPerMeter	+2Ah	4	Разрешение по Y в пикселах на метр
biClrUsed	+2Eh	4	Число используемых цветов
biClrImportant	+32h	4	0, если число цветов максимально

Групповое кодирование. Формат PCX

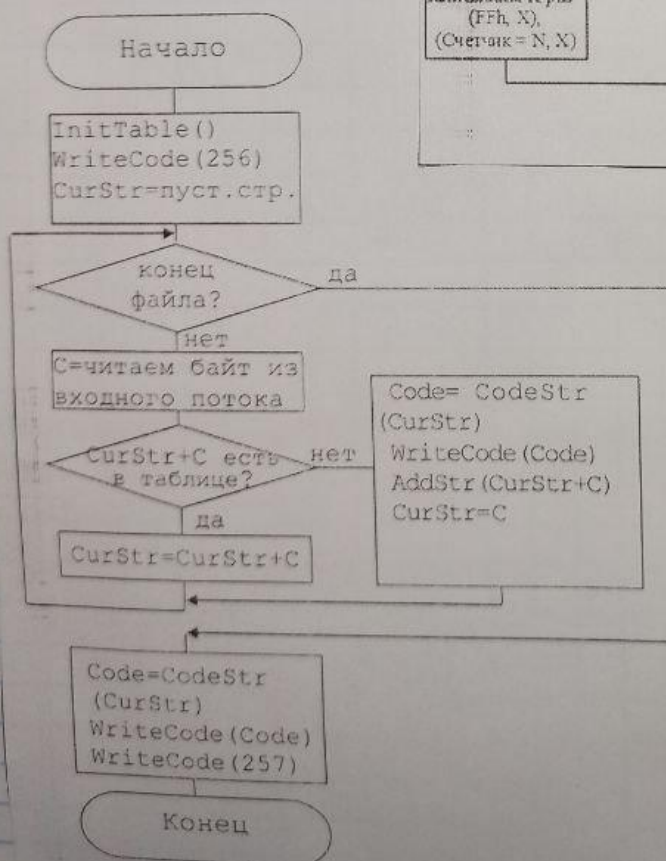
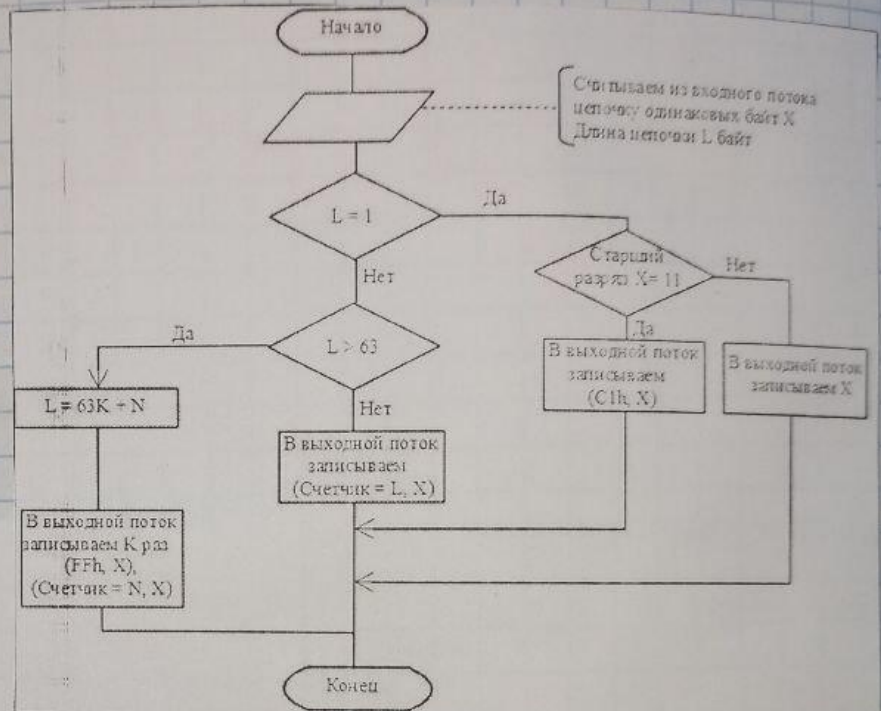
RLE – run length encoding



Основные принципы работы алгоритма RLE:

- данные кодируются побайтно;
- формат байта-сметки количества повторов следующий: 11xxxxxx. П.с. для отметки используются 2 старших разряда, они устанавливаются в 11₂, а оставшиеся 6 младших разрядов хранят количество повторов;
- байт остаточной информации со старшими битами, отличными от 11₂, кодируется в выходной поток без изменений.

• Байт несжатой информации со старшими битами 112 подгружается в виде пары (номер сессии повторов = 1, значение)



← Метод сжатия LZW
Форматы GIF и TIFF

Рис. 4.3. Сжатие данных по методу LZW

Пример. Сумма номеров:

10h, COh, ..., COh (145 pag), 56h, 28h, 7Ah, ..., 7Ah (54 page)

$$10h = \underline{0001,0000},$$

$$145 = 65 \cdot 2 + 15$$

$$\begin{array}{r} 1101 \mid 0011 \\ 2 \mid 3 \end{array}$$

Box: 10h, (FFh, COh), (EFh, COh), (23h, COh), 56h,
(Ch, 28h), (F6h, 7Ah)

$$56h = 01010110$$

$$28h = 11011000_2$$

$$54 = 16 \cdot 3 + 6$$

$$\begin{array}{r} 1111,0110 \\ F \quad 6 \end{array}$$

$$\frac{202}{12} = 16,8$$

W. 45h, 11h, ..., 11h, 7Ah, F0h, 85h, ..., 85h
111 85

$$F = 45h = \underline{0100}, \underline{0101},$$

$$111 = 63 + 48$$

FFh

$$48 = 32 + 16$$

$$\begin{array}{r} 1111,0000 \\ F \quad 0 \end{array}$$

$$7Ah = \underline{0111}, \underline{1010}$$

F0h (X=H)

$$85 = 63 + 16 + 4 + 2$$

$$\begin{array}{r} 1101,0110 \\ D \quad 6 \quad h \end{array}$$

Boxes: 45h, (FFh, 11h), (F0h, 11h), 7Ah, (Ch, F0h),

(FFh, 85h), (2Ch, 85h)

Декодирование:

$$45h = 01000101$$

01 ≠ 11 ⇒ число 45h

$$FF = 11111111$$

11 ⇒ первая часть информации

$$F0h = 11110000$$

48 битов

$$D3h \dots D3h \quad 55h \quad F3h \quad 07h$$

$$30 = 2^4 + 2^3 + 2 = 011110$$

4 3 2 1 0
10 + 8 + 4 + 2

$$\frac{1101}{D} \frac{1110}{E}$$

$$DEh \quad D3h \quad 55h \quad Ch \quad F3h \quad 07h$$

$$55h = 01010101$$

$$F3h = 11110011$$

$$07h = 00000111$$

символ 11

Методы сжатия LZW. Форматы GIF и TIFF.

Init Table () - инициализация таблицы и помещается в нее символы основной гаммы
символы 256 (0...255)

Clear Code - код очистки таблицы (256)

Code End of Information - код конца информации

Code End of File - 257