Принципы сжатия видеоинформации.

Лекция 2. ***Сжатие по алгоритму JPEG***

9 семестр, кафедра РТПи АС, лектор:
доцент, к.т.н. Бугаев Юрий Николаевич
и
д.т.н. Дворкович Александр Викторович

2016 г.

Конвейер операций, используемый в алгоритме JPEG

Шаги преобразования

- **1 шаг.** Преобразования цветового пространства в сигнал яркости Y и два цветоразностных сигнала U и V

  Упрощенно перевод из цветового пространства RGB в цветовое пространство YCrCb можно представить так:

$$\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.4187 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix}$$

- Обратное преобразование осуществляется умножением вектора YUV на обратную матрицу

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{vmatrix} * \left(\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} - \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \right)$$

1 шаг. Преобразования цветового пространства в сигнал яркости Y и два цветоразностных сигнала U и V.

- Преобразования цветового пространства в сигнал яркости **Y** и два цветоразностных сигнала **U** и **V**.
- Хотя в принципе сам стандарт этого не требует, но это позволяет повысить эффективность сжатия.
- Степень сжатия компоненты яркости будет меньше, чем цветоразностных компонент, так как человеческий глаз меньше чувствителен к изменения цвета и значительно более чувствителен к изменению яркости. изменению яркости.

2 шаг. Прореживание U и V данных цветности

Прореживание **U** и **V** данных цветности. При прореживании отбрасываются цветоразностные компоненты строк или столбцов пикселов с определенными номерами (например, каждой второй строки или каждого второго столбца)

Формируем из каждой три рабочие матрицы ДКП — по 8 бит отдельно для каждой

При этом мы теряем 3/4 полезной информации о цветовых составляющих изображения и получаем сразу сжатие в два раза.

3 шаг. Преобразование блоков изображения при помощи двухмерного ДКП

- Преобразование небольших блоков изображения при помощи двухмерного дискретного косинусного преобразователя. Обработка ведется блоками 8 х 8 пикселей, т.е. обрабатывается сразу 64 пикселя. Выбор такого блока обусловлен двумя причинами
- **Важнейшей особенностью этой матрицы является т о , что основную энергию несут первые ее коэффициенты, а энергия последующих быстро убывает.**
- В упрощенном виде это преобразование можно представить так:

$$Y[u,v] = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C(i,u) \times C(j,v) \times \quad y[i,j]$$

где $$C(i,u) = A(u) \times \quad \cos\left(\frac{(2\times i+1)\times u\times \pi}{2\cdot n}\right)$$

$$A(u) = \begin{cases} \frac{1}{\sqrt{2}}, & \text{for} \quad u \equiv 0 \\ 1, & \text{for} \quad u \neq 0 \end{cases}$$

4 шаг. Операция квантования

- Матрица проходит операцию квантования, которая позволяет сократить разрядность коэффициентов.
- Это математически соответствует делению матрицы коэффициентов дискретного косинусного преобразования размерностью 8 x 8 на матрицу квантования также размерностью 8 x 8 .

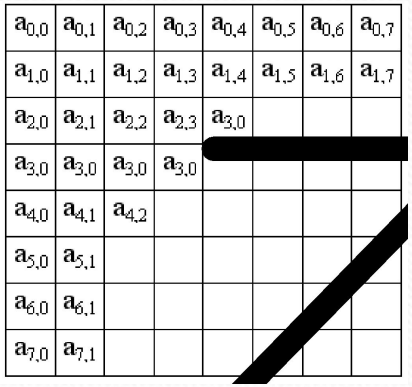
$$Yq[u,v] = IntegerRound\left(\frac{Y[u,v]}{q[u,v]}\right)$$

- матрица квантования q[u,v] (далее МК).

- После квантования значения чисел в левом верхнем углу становятся значительно меньше , а ближе к правому нижнему углу становятся равными нулю. Именно в этой операции происходит основная и необратимая потеря информации.
- Яркостная компонента квантуется обычно с большим числом разрядов, чем цветоразностные

5 шаг. Матрица вытягивается в строку данных.

- Матрица после квантования вытягивается в строку данных так, что все последовательности нулей правого нижнего угла оказываются в конце строки.
- В некоторых версиях информация о яркости и цвете кодируется так, что сохраняются только отличия между соседними блоками.
- Переводим матрицу 8x8 в 64-элементный вектор при помощи “зигзаг”-сканирования, т.е. берем элементы с индексами (0,0), (0,1), (1,0), (2,0)...



- Таким образом, в начале вектора мы получаем коэффициенты матрицы, соответствующие низким частотам, а в конце — высоким.

6 Шаг. Статистическое кодирование по методу Хаффмана

- Статистическое кодирование по методу Хаффмана, считается, что этот метод сжимает без потерь. Сначала анализируется вся последовательность символов. Часто повторяющимся сериям бит присваивается короткие элементы (маркеры) . В частности последние нули в конце строки могут быть заменены одним символов конца строки. Так как все блоки имеют точно известную и одинаковую длину, то всегда можно точно определить, сколько нулей было опущено.

Положительными сторонами алгоритма является то, что:

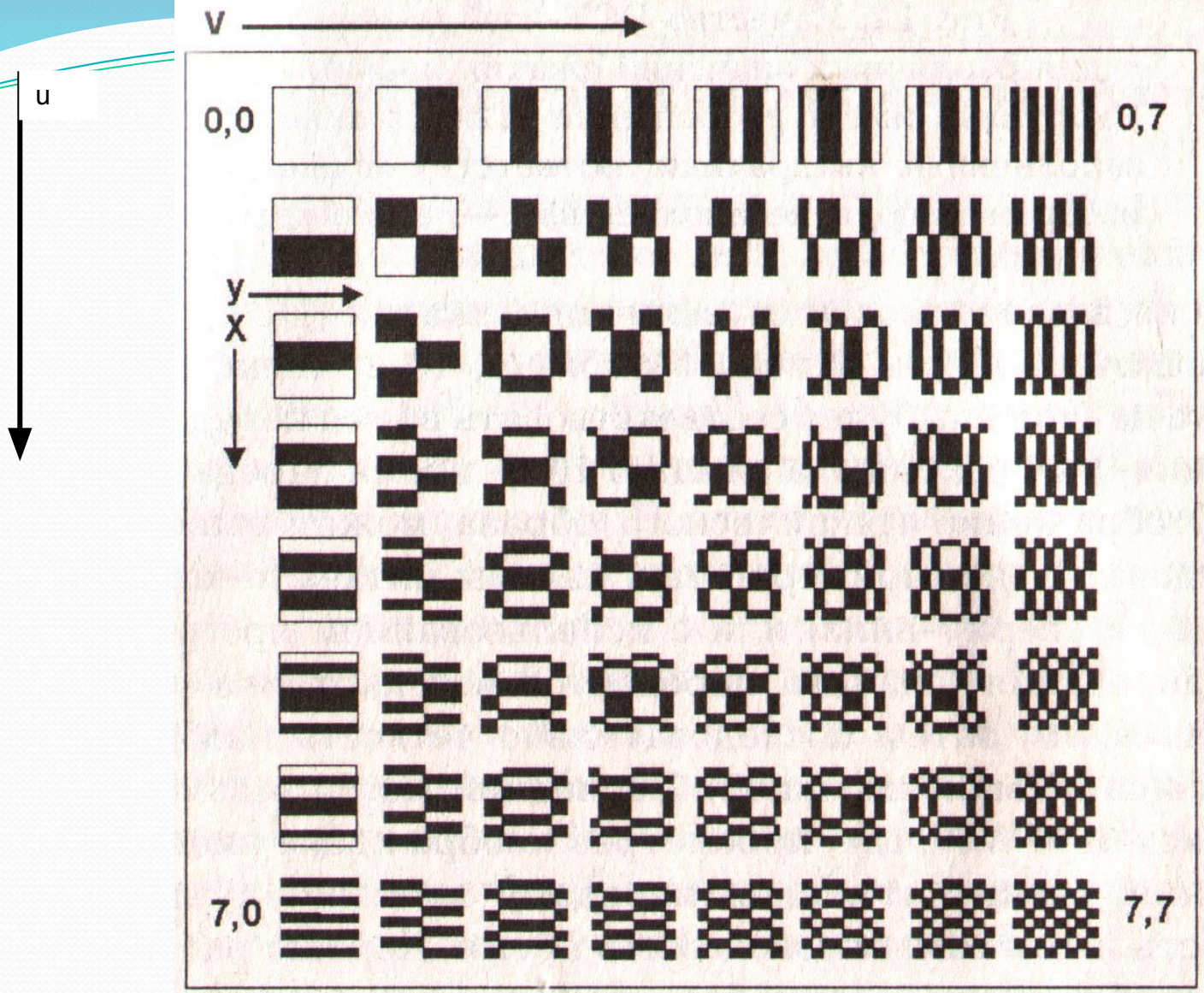
- **Задается степень сжатия.**
- **Выходное цветное изображение может иметь 24 бита на точку.**

Отрицательными сторонами алгоритма является то, что:

- **При повышении степени сжатия изображение распадается на отдельные квадраты (8x8).**
- **Проявляется эффект Гиббса.**

Характеристики алгоритма JPEG:

- **Коэффициенты компрессии: 2-200 (Задается пользователем).**
- **Класс изображений: Полноцветные 24 битные изображения, или изображения в градациях серого без резких переходов цветов (фотографии).**
- **Симметричность: 1**



Графическое представление двумерного преобразования

Алгоритм Хаффмана

Классический алгоритм Хаффмана.

- Алгоритм использует только частоту появления одинаковых байт в изображении. Сопоставляет символам входного потока, цепочку бит меньшей емкости и наоборот встречающимся редко цепочку большой длины. Для сбора статистики требует двух проходов по изображению.

Определения

Определение 1

- Пусть задан алфавит ψ = {a1, ……ar}, состоящий из конечного числа букв. Конечную последовательность символов из ψ
- A = a i1, ai2,…..ain (1.4)
- Будем называть словом в алфавите ψ, а число n - длина слова A, длина слова обозначается ***как l (A).***
- Пусть задан алфавит Ω, Ω = {b1,….bq}. Через B обозначим слово в алфавите Ω и через S(.Ω) –множество непустых слов в алфавите Ω.
- Пусть S = S (ψ) множество всех непустых слов в алфавите ***ψ, S'*** – некоторое подмножество множества S. Пусть задано отображение F, которое каждому слову A, A s(ψ) ставит в соответствие слово

  B = F(A), B S(Ω) (1.5)
- Слово B будем называть кодом сообщения A, а переход слова A к его коду-кодированием.

Определение 2.

Рассмотрим *соответствие между буквами алфавита* Y *и некоторыми словами алфавита* Ω:

$$\begin{array}{l} a1 \text{ -- } B\ 1; \\ a2 \text{ -- } B\ 2; \\ \_\_\_\_\_ \\ ar \text{ -- } B\ r; \end{array} \qquad (1.6)$$

Это соответствие называют ***схемой*** и обозначают $\sum$. Оно определяет кодирование следующим образом:

- каждому слову $A = ai1, ai2, \ldots, ain$ из $S'(\psi) = s(\psi)$ ставится в соответствие слово $B = Bi1, Bi2, \ldots, Bin$, ***называемым кодом слова*** A. Слова $B1, \ldots Br$ называются элементарными кодами. Такой вид кодирования называется ***алфавитным кодированием***.

Определение 3.

Пусть слово B имеет вид

$$B = B'B'' \qquad (1.7)$$

Тогда слово B’ называется началом или префиксом слова B, а слово B” – концом слова B. При этом пустое слово L и само слово B считается началом и концом слова B.

Определение 4.

Схема S обладает свойством префикса,
если для любых i и j ($1 \le i,\ j \le r$, $i \neq j$) слово Bi не является префиксом слова Bj.

Теорема 1. Если схема $\sum$ обладает свойством префикса то алфавитное кодирование будет взаимно однозначным.

Предположим, что задан алфавит $\psi = \{a1,\ldots,ar\}$ $(r > 1)$ и набор вероятностей $p1,\ldots,pr$ $(=1)$ появления символов $a1,\ldots,ar$. Пусть далее, задан алфавит W, $W = \{b1,\ldots,bq\}$ где $(q > 1)$.

Тогда можно построить целый ряд схем алфавитного кодирования.

$$\begin{matrix} a1 & - & B1 \\ \ldots & & (1.8) \\ ar & - & Br \end{matrix}$$

обладающих свойством взаимной однозначности.

Для каждой схемы можно ввести среднюю длину l ср, определяемую как мат. ожидание длины элементарного кода:

$$lcp = \ , \qquad = l\,(Bi) \text{ – длины слов} \qquad (1.9)$$

Длина lcp, показывает, во сколько раз увеличивается средняя длина при кодировании со схемой Σ

Можно показать, что lcp достигает величины своего минимума l* на некоторой схеме Σ и определена как

$$l^* = \min_{\Sigma} l_{cp}^{\Sigma}$$

Определение 5.

- Коды, определяемые схемой с $lcp = l^*$, называются кодами с минимальной избыточностью или кодами Хаффмана. Коды с минимальной избыточностью дают в среднем минимальное увеличение длин слов при соответствующем кодировании.
- В нашем случае алфавит $= \{a1,\ldots,ar\}$ задает символы входного потока, а алфавит $\{0,1\}$, т.е. состоит всего из нуля и единицы.

Алгоритм построения схемы

- ***Шаг 1.***

Упорядочиваем все буквы входного алфавита в порядке убывания вероятности. Считаем все соответствующие слова Bi , из алфавита Ω = {0,1} пустыми.

Шаг 2.

Объединением два символа air-1 и a ir с наименьшими вероятностями pir-1 и p ir в псевдосимвол a’ {air-1, a ir} с вероятностью pir-1 + p ir. Дописываем 0 в начало слова Bir-1 (Bir-1 = 0, Bir-1) и 1 в начало слова Bir (Bir = 1, Bir).

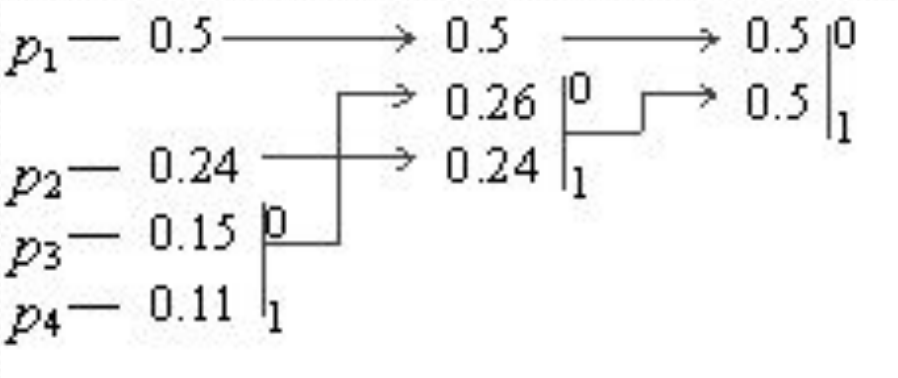
Шаг 3.

Удаляем из списка упорядоченных символов air-1 и a ir и заносим туда псевдосимвол a’ {air-1, a ir}. Проводим шаг 2, добавляя при необходимости 1 или 0 до тех пор, пока в списке не останется один псевдосимвол.

Пример:

- Пусть у нас есть 4 буквы в алфавите $\Psi = \{a_1, a_r\}$ $(r=4)$, $\sum_{i=1}^{4} pi = 1$
- p1= 0.5; p2 = 0,24; p3 = 0,15; p4 = 0,11 ( ).

Процесс построения схемы можно представить так:



Производя действия, соответствующие 2-му шагу, мы получаем псевдосимвол с вероятностью 0.26 (и приписываем 0 и 1 соответствующим словам). Повторяя же эти действия для измененного списка, мы получаем псевдосимвол с вероятностью 0.5. И, наконец, на последнем этапе мы получаем суммарную вероятность 1.

Для того, чтобы восстановить кодирующие слова, нам надо пройти по стрелкам от начальных символов к концу получившегося бинарного дерева. Так, для символа с вероятностью p4, получим B4=101, для p3 получим B3=100, для p2 получим B2=11, для p1 получим B1=0. Что означает схему:

a1 — 0,
a2 — 11
a3 — 100
a4 — 101

Эта схема представляет собой префиксный код, являющийся кодом Хаффмана. Самый часто встречающийся в потоке символ a1 мы будем кодировать самым коротким словом 0, а самый редко встречающийся a4 длинным словом 101. Для последовательности из 100 символов, в которой символ a1 встретится 50 раз, символ a2 — 24 раза, символ a3 — 15 раз, а символ a4 — 11 раз, данный код позволит получить последовательность из 176 бит (). Т.е. в среднем мы потратим 1.76 бита на символ потока.

Характеристики классического алгоритма Хаффмана:

- **Коэффициенты компрессии**: 8, 1,5, 1 (Лучший, средний, худший коэффициенты).
- **Класс изображений**: Практически не применяется к изображениям в чистом виде. Обычно используется как один из этапов компрессии в более сложных схемах.
- **Симметричность:** 2 (за счет того, что требует двух проходов по массиву сжимаемых данных). (время разархивации не равно времени архивации).
- **Характерные особенности:** Единственный алгоритм, который не увеличивает размера исходных данных в худшем случае (если не считать необходимости хранить таблицу перекодировки вместе с файлом).

Векторное квантование

- **Считается перспективным и используется в JPEG векторное квантование.**
- **Векторное квантование эффективно, когда требуемое число битов на элемент изображения должно быть меньше одной двоичной единицы.**
- **Векторный квантователь состоит из множества, называемого кодовой книгой, содержащей L кодовых векторов размерностью K. Векторы формируются путем деления исходного изображения на смежные неперекрывающиеся блоки изображений. Если кодовая книга создана, и она имеется на приемной и передающей стороне, то при получении номера индекса вектора приемник выбирает из своей книги соответствующий вектор и заполняет им необходимое место на изображении.**
- **Векторное квантование является очень экономным. Почти все первые программы САПР, которые строились на ЭВМ с ограниченными возможностями, имели векторную структуру представления данных.**

Статистическое (энтропийное) кодирование (кодирование по Шеннону-Фэно)

- Статистическое (энтропийное) кодирование используется для уменьшения избыточности сообщений, обусловленной неравной вероятностью появления элементов. Часто встречающиеся высоковероятные элементы кодируются короткими кодовыми комбинациями, а редко встречающиеся маловероятные можно кодировать более длинными кодовыми комбинациями. Необходимо также, чтобы короткие кодовые комбинации не совпадали с началом более длинных. В противном случае при декодировании возникнут ошибки.
- Подлежащие кодированию элементы располагаются в первом столбце таблицы в порядке убывания их вероятности.
- Элементы сообщений разбиваются на две группы с примерно равными суммарными вероятностями. Элементам первой группы в качестве первого знака присваивается 0, элементам второй -1.
- Элементы, входящие в каждую группу вновь разбиваются на две группы. Элементам первой группы присваивается второй индекс 0, второй группы -1.
- Этот процесс продолжается пока в каждом элементе не останется по одному элементу.