UNIVERSITATEA TEHNICĂ A MOLDOVEI
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Informatică și Ingineria Sistemelor

GRAFICA PE CALCULATOR

ТЕМА 5. СЖАТИЕ ИЗОБРАЖЕНИЙ

l.u., dr. NASTAS Andrei

СОДЕРЖАНИЕ

5.5. Сжатие JPEG (Объединенная группа экспертов в области фотографии)

Область сжатия (кодирования) изображений связана с минимизацией количества битов, необходимых для восстановления изображения, посредством приложений, также в передаче и хранении изображений. Применение в области *передачи изображений* находят в: радиовещании, космической связи, радиолокационных и гидролокационных, телекоммуникационных сетях, факсимильных передачах, телеконференциях и т.д. Сжатие изображений имеет важное значение с точки зрения *хранения изображений*: в приложениях медицинской визуализации, в цифровых видеотехнологиях, для создания мультимедийных документов и т. Д. Новые технологии сжатия изображений обеспечивают интеграцию в приложения цифровых изображений и видео. Степень сжатия теперь достигла до 1:100, в зависимости от качества восстановленного изображения. Техники сжатия недостаточно, чтобы иметь возможность решать проблемы, возникающие в мультимедийных приложениях. Для того чтобы иметь возможность достичь переносимости цифровых видеоизображений и последовательностей приложений на нескольких системах, необходимо внедрить стандарты сжатия мультимедийных данных. Эти стандарты устанавливают порядок хранения и передачи сжатых данных с целью их возможного использования. Наиболее широко используемым стандартом сжатия статических изображений является стандарт JPEG, созданный Объединенной группой экспертов в области фотографии. Метод сжатия относится к типу «с потерями», будучи разработанным таким образом, чтобы использовать ограничения в восприятии видео человеческим глазом. Этот стандарт позволяет установить степень качества/сжатия и работает с одинаковыми цветовыми уровнями, насчитывающими 24 (16,7 млн цветов), независимо от общего количества цветов в изображении. На данный момент это один из самых распространенных форматов графических файлов.

5.5. Сжатие JPEG

Формат JPEG рекомендуется для отображения изображений, визуализированных с широким спектром цветов или для изображений фотографической точности. JPEG использует метод переменного сжатия, что приводит к получению очень маленьких файлов по сравнению с другими форматами.

Стандарт JPEG основан на преобразовании первичной информации из временной области в частотную. Известно, что изображения сильно коррелируют пространственно, то есть пиксель изображения также содержит информацию о соседних пикселях. Пространственная корреляция, характеризующая изображения, представляет собой избыточность с информационной точки зрения и уменьшается математическими преобразованиями, которые играют роль концентрации энергии изображения в как можно меньшем количестве элементов. Математические преобразования от времени к частоте сами по себе не представляют собой сжатие данных. Только последующие операции, а именно квантование и энтропическое кодирование, представляют собой сжатие данных. Уменьшение пространственной избыточности выполняется как для исходного изображения, так и для остаточной погрешности, как будет показано дальше. При восстановлении изображений, после того, как они были сжаты JPEG, объем информации меньше исходного, без видимого ухудшения качества. При преобразовании изображения из временной области (пикселей) в частотном диапазоне сохраняются только низкочастотные компоненты изображения. Высокочастотные компоненты могут быть уменьшены, без беспокоящего нарушения зрительного восприятия изображения. Очевидно, что это определяется принятой степенью сжатия.

5.5. Сжатие JPEG

JPEG - это сложный метод сжатия с потерями для цветных или черно-белых изображений (в оттенках серого). Преимущество JPEG заключается в том, что он использует множество параметров, что позволяет пользователю регулировать объем потерянных данных, а также степень сжатия. В настоящее время это лучший стандарт для сжатия неподвижных изображений. Стандарт реализован как в программном, так и в аппаратном формате для удовлетворения потребностей обработки мультимедийных приложений в режиме реального времени. Первоначально созданный для сжатия неподвижных изображений, стандарт был распространен и на видеоряды. Стандарт, созданный для видеопоследовательностей, называется M-JPEG (Motion JPEG). В основном в случае цифровых видеопоследовательностей каждый кадр рассматривается как фиксированное изображение и сжимается со стандартом JPEG. Метод не самый эффективный с точки зрения размера сжатия, но он предлагает альтернативу для сжатия цифрового видео. Зачастую человеческий глаз не различает никакой деградации изображения даже при степени сжатия 10:1 или 20:1.

5.5. Сжатие JPEG

Существует четыре основных режима работы, определенных стандартом JPEG:

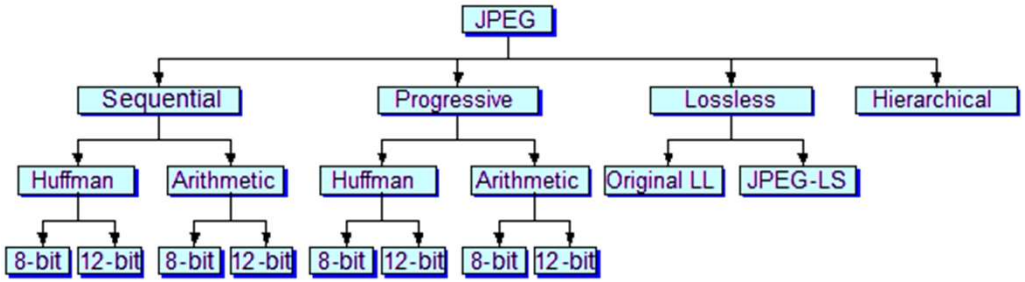
— базовый режим, в котором каждый компонент изображения кодируется одним сканированием влево-вправо, соответственно вверх-вниз;

– DCT расширенное кодирование с потерями, при котором выполняется прогрессивное кодирование входных спектров изображения;

— кодирование без потерь, при котором изображение кодируется таким образом, что при декодировании гарантируется точное воспроизведение;

– иерархическая кодировка, при которой изображение кодируется в нескольких разрешениях.

Далее подробно будет представлен только первый режим работы, для других только суть.



5.5.1. Базовый режим (baseline)

Блочная схема алгоритма кодирования JPEG показана на рисунке 5.8.

Принцип сжатия методами типа DCT.

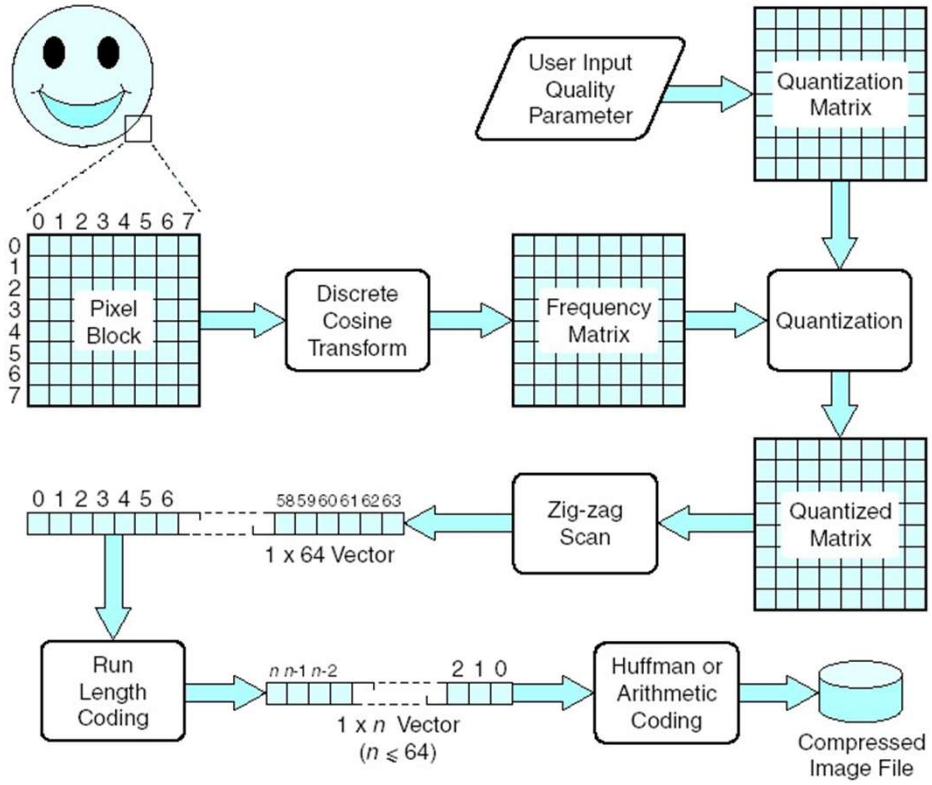


Рис. 5.8. Алгоритм кодирования JPEG

5.5.1. Базовый режим

Сжатие с потерями включает в себя несколько этапов обработки, а именно:

- **Преобразование из представления (R,G,B) в представление (Y,U,V)**

В процедурах сжатия изображения предпочтительно иное цветовое представление, отличное от обычного (R,G,B), а именно представление (Y,U,V), полученное с помощью формул 5.1. Значения компонентов Y, U и V находятся в диапазоне от -128 до 127. Полезность этого эквивалентного представления показана на рисунке 5.9, на котором видны разложения в виде (R,G,B) соответственно (Y,U,V) одного изображения.

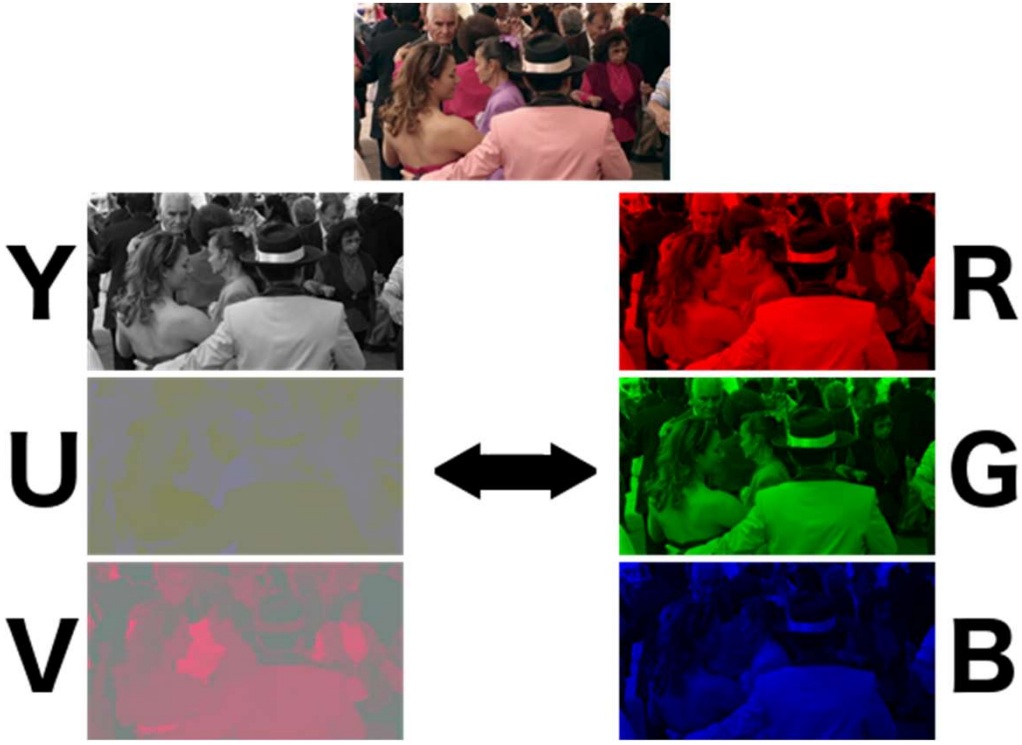


Рис. 5.9. Компоненты R, G, B, Y, U, V одного изображения

5.5.1. Базовый режим

Анализируя этот пример, можно сделать некоторые важные наблюдения, а именно:

– компонент Y соответствует черно-белому изображению;

— компоненты U и V содержат гораздо меньше деталей и представляют гораздо меньшую контрастность.

Из-за отсутствия деталей и низкой контрастности компонентов U и V, они подвержены недостаточной дискретизация 2-х мерным фактором в обоих направлениях, вертикальном и горизонтальном, с учетом того, что восприятие глазом ниже при сигналах цветности, по сравнению с яркостью. Способ выполнения недостаточной дискретизация (андерсамплинга) заключается в замене блоков из 2x2 точек на одну точку, которая имеет интенсивность, равную среднему значению этих 4 точек. В этих условиях изображение будет описано компонентами U' и V’, как показано на рисунке 5.10.

При этих операциях достигается первое сжатие, с коэффициентом 2:1. Таким образом, для представления (R, G, B) для изображения в примере с разрешением 320 x 240 точек требуется 3 компонента по 320 x 240 = 76800 элементов, то есть всего 230400 элементов (байтов). Представление (Y, U', V') требует 320 x 240 = 76800 элементов для Y и 160 x 120 = 19200 элементов для U' и V', т.е. в общей сложности 115200 элементов (байтов).

Компонент U' (недостаточная дискретизация)

Компонент V' (недостаточная дискретизация)

Рис. 5.10. Недостаточно дискретизированные компоненты U' и V'

5.5.1. Базовый режим

• **Разложение на блоки**

Процедура сжатия применяется к блокам изображений размером 8x8 точек. Если ни одно из измерений изображения не кратно 8, кодировщик копирует последний столбец или строку до тех пор, пока окончательная длина не будет кратна 8. Эти дополнительные строки или столбцы удаляются в процессе декодирования.

Три компонента Y, U' и V' разлагаются на блоки размером 8x8. Из-за низкого разрешения, после недостаточной дискретизации, следует, что на 4 (2x2) блока компоненты Y соответствует одному блоку компонентов U', соответственно V'.

В случае формата JPEG три компонента блоков изображения переплетаются. Для нумерации блоков, согласно рисунку 5.11, порядок их обработки будет Y1, Y2, Y3, Y4, U1, V1, Y5, Y6, Y7, Y8, U2, V2,....:

Каждый блок обрабатывается с использованием одной и той же процедуры.

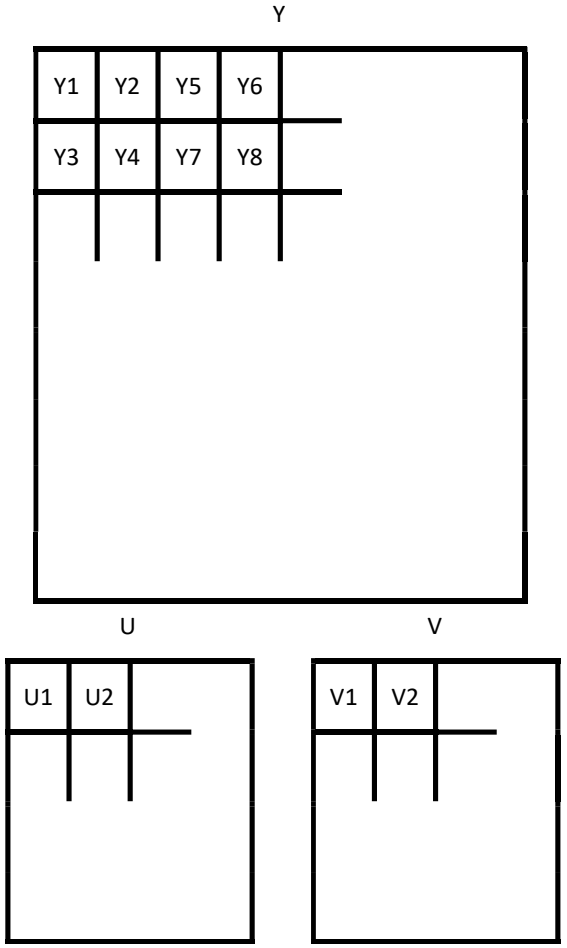


Рис. 5.11. Порядок обработки блоков

5.5.1. Базовый режим

- **Применение дискретного косинусного преобразования**

Исходные значения компонентов Y, U', V' содержатся в диапазоне [0, 2b–1], где b представляет собой количество битов/выборку. Эти значения смещены в диапазоне [–2*b*–1, 2*b*–1 –1], центрированном относительно нуля, чтобы достичь большей вычислительной точности при применении DCT (Discrete Cosine Transform). Для первого уровня кодировки JPEG *b* = 8, так что исходные значения, содержащиеся в диапазоне [0, 255], перемещаются в диапазон [–128, 127]. Затем каждый компонент разделяется на блоки размером 8 x 8 пикселей, как показано на рисунке 5.8. К каждому полученному таким образом блоку применяется дискретное двумерное косинусное преобразование, использующее уравнения (5.36), (5.37), (5.38):

Чтобы проиллюстрировать, как работает алгоритм, будет использоваться блок изображения размером 8 × 8 размеров, как показано в таблице 5.4.

$$\text{DCT: } G_{ij} = \frac{1}{4} C_i C_j \sum_{x=0}^{7} \sum_{y=0}^{7} p_{xy} \cos\left(\frac{(2y+1)j\pi}{16}\right) \cos\left(\frac{(2x+1)i\pi}{16}\right) \quad (5.36)$$

$$\text{IDCT: } p_{xy} = \frac{1}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} C_i C_j G_{ij} \cos\left(\frac{(2y+1)j\pi}{16}\right) \cos\left(\frac{(2x+1)i\pi}{16}\right) \quad (5.37)$$

unde

$$C_i = C_j = \frac{1}{\sqrt{2}}, \quad \text{dacă } i = j = 0$$
$$C_i = C_j = 1, \quad \text{în rest} \quad (5.38)$$

Таблица 5.4. 8 x 8 размеров блока изображения

124	125	122	120	122	119	117	118
121	121	120	119	119	120	120	118
126	124	123	122	121	121	120	120
124	124	125	125	126	125	124	124
127	127	128	129	130	128	127	125
143	142	143	142	140	139	139	139
150	148	152	152	152	152	150	151
156	159	158	155	158	158	157	156

5.5.1. Базовый режим

Рассматривая блок 8 × 8 пикселей в таблице 5.4, вычитается 128 из каждого элемента и применяя DCT, получаются коэффициенты DCT, показанные в таблице 5.5.

Видно, что элементы этой матрицы имеют большие значения, сконцентрированные в левом верхнем углу, остальные – небольшие, почти нулевые значения. Объяснение этому явлению дает тот факт, что дискретное косинусное преобразование достигает разложения «по частоте». Таким образом, коэффициенты в левом верхнем углу соответствуют низким частотам - медленным колебаниям интенсивности между пикселями, а по мере продвижения в правый нижний угол коэффициенты соответствуют высоким частотам - быстрым колебаниям интенсивности, задаваемым мелкими деталями на изображении. В целом, в реальном изображении высокие частоты имеют меньший вес, чем низкие частоты, что объясняет значения, полученные от преобразования. Эта ситуация проиллюстрирована на рисунке 5.12

Таблица 5.5. Матрица G коэффициентов DCT, соответствующая блоку данных изображения после смещения

39,88	6,56	-2,24	1,22	-0,37	-1,08	0,79	1,13
-102,43	4,56	2,26	1,12	0,35	-0,63	-1,05	-0,48
37,77	1,31	1,77	0,25	-1,50	-2,21	-0,10	0,23
-5,67	2,24	-1,32	-0,81	1,41	0,22	-0,13	0,17
-3,37	-0,74	-1,75	0,77	-0,62	-2,65	-1,30	0,76
5,98	-0,13	-0,45	-0,77	1,99	-0,26	1,46	0,00
3,97	5,52	2,39	-0,55	-0,051	-0,84	-0,52	-0,13
-3,43	0,51	-1,07	0,87	0,96	0,09	0,33	0,01

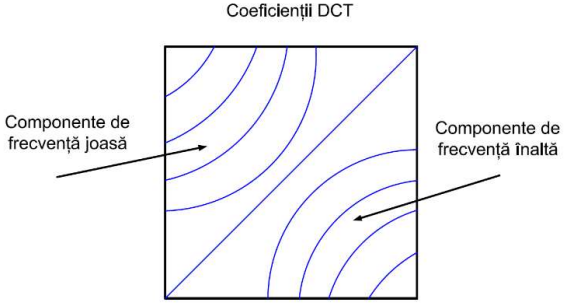


Рис. 5.12. Распределение частотных компонентов в матрице квантования коэффициентов DCT преобразованной матрицы

5.5.1. Базовый режим

- **Квантование преобразованной матрицы**

Операция квантования является единственной, в которой происходит потеря информации. Алгоритм JPEG использует коэффициенты дизеринга для квантования различных входных коэффициентов. Размер шага квантования организован в таблицу, называемую таблицей квантования. Пример таблицы квантования в процессе сжатия JPEG приведен в таблице 5.6. Каждое количественное значение представлено меткой. Квантование означает деление элемента по элементам матрицы *G* с матрицей квантования *Q*, с сохранением только целой части, в результате чего получается матрица *I*.

Метка, соответствующая количественному значению коэффициентов G_{ij} преобразования, равна

$$I_{i,j} = \left\lfloor \frac{G_{ij}}{Q_{ij}^{t}} + 0,5 \right\rfloor \qquad (5.37)$$

где $Q_{ij}{}^{t}$ — элемент (i, j) таблицы квантования, а [*x*] — наибольшее целое число меньше x. Рассмотрим коэффициент G_{00} в таблице 5.5, который равен 39, 88. Из таблицы 5.6, а, $Q_{00}{}^{t}$ равно 16, так

$$I_{00} = \left\lfloor \frac{39,88}{16} + 0,5 \right\rfloor = \lfloor 2,9925 \rfloor = 2 \qquad (5.38)$$

Таблица 5.6. Коэффициенты квантования для яркости (a) и для цветности (b)

a)

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

b)

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

5.5.1. Базовый режим

Перестроенное значение получается из метки путем умножения его на соответствующую запись в таблице квантования. Таким образом, перестроенное значение ϑ_{00} будет I_{00} x $Q_{00}{}^{t}$, т.е. 2 x 16 = 32. Погрешность квантования в данном случае составляет 39, 88 – 32 = 7,88. Аналогичным образом, из таблиц 5.5 и 5.6 G_{01} равен 6, 56 и $Q_{01}{}^{t}$ равен 11, поэтому

$$I_{01} = \left\lfloor \frac{6,56}{11} + 0,5 \right\rfloor = \lfloor 1,096 \rfloor = 1 \qquad (5.39)$$

Перестроенное значение равно 11, а ошибка квантования равна 11 – 6,56 = 4,44. Продолжая таким образом, получаются образцы, приведенные в таблице 5.7.

Замечено, что после квантования в матрице квантованных коэффициентов гораздо более выражено явление концентрации больших значений в левом верхнем углу, а преобладания малых значений (даже нуля) в остальных.

Таблица 5.7. Количественные коэффициенты, полученные с помощью таблицы квантования коэффициентов

2	1	0	0	0	0	0	0
-9	0	0	0	0	0	0	0
3	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	0

5.5.1. Базовый режим

Потеря информации обусловлена достижением деления с сохранением только целой части результатов. Таким образом, значения теряют точность (маленькие превращаются в ноль). Эффект этой потери, однако, не заметен, поскольку, как мы видим, наибольшие потери концентрируются на уровне высокочастотных коэффициентов, которые имеют небольшой вес в изображении и которые, соответствуют мелким деталям, менее заметными человеческим глазом.

Роль операции квантования заключается в получении как можно большего количества нулевых или низких значений, которые имеют преимущество эффективного кодирования, выполненного позже. Дискретное косинусное преобразование дает возможность выполнить эту операцию так, чтобы эффект потери информации был как можно меньше.

Из таблицы 5.7, содержащей квантованные образцы, видно, что размер шага увеличивается по мере удаления от коэффициента постоянного тока. Поскольку ошибка квантования является возрастающей функцией размера шага, на высокочастотные коэффициенты будет влиять более высокая погрешность квантования, чем низкочастотные. Решение об относительном размере шага квантования основывается на способе восприятия ошибок зрительной системы человека. Различные коэффициенты трансформации имеют разную рецептивную значимость. Ошибки квантования в низкочастотных коэффициентах постоянного и переменного тока легче обнаружить, чем ошибки квантования для высокочастотных коэффициентов переменного тока. Именно поэтому более высокая ступень используется для менее важных коэффициентов восприятия.

5.5.1. Базовый режим

Поскольку квантизаторы имеют уровень 0 в качестве уровня реконструкции, процесс квантования также функционирует как пороговая операция кодирования. Все коэффициенты с амплитудой менее половины размера шага будут равны нулю. Поскольку размер шага в конце зигзагообразного сканирования высок, вероятность нахождения длинной последовательности нуля увеличивается. Этот эффект может быть способом изменения курса. Увеличивая шаг, можно уменьшить количество различных нулевых значений, требуемых для передачи, что означает уменьшение количества требуемых битов.

5.5.1. Базовый режим

• **Кодирование элементов в матрице количественных коэффициентов**

Квантованные коэффициенты сканируются зигзагами с целью получения одномерной последовательности, которая будет применена энтропийному энкодеру. Как уже упоминалось ранее, первый коэффициент называется DC (постоянного тока), и представляет собой среднее значение интенсивности блока. Остальные 63 коэффициента называются коэффициентами переменного тока (AC coefficients). Зигзагообразное сканирование осуществляется в представлении упорядочивания по частотному спектру. Поскольку высокочастотные компоненты имеют приблизительно нулевые значения, сканирование зигзагом приводит к строке нулей в конце последовательности, что позволяет эффективно кодировать при помощи RLC (Run Length Coding) и Хаффмана.

В алгоритме сжатия JPEG используются две различные процедуры кодирования. Первая процедура используется для кодирования элемента I_{00}, который является коэффициентом постоянного тока, вторая процедура используется для кодирования других 63 коэффициентов переменного тока. Коэффициент постоянного тока дифференциально кодируется из коэффициента постоянного тока в предыдущем блоке с использованием алгоритма DPCM (Differential Pulse Code Modulation). Коэффициенты переменного тока кодируются RLC. Коэффициенты постоянного и переменного тока, закодированные таким образом, будут затем энтропически закодированы с использованием кодирования Хаффмана.

5.5.1. Базовый режим

Кодирование коэффициентов постоянного тока

На рисунке 5.13 показано, что базовая матрица, соответствующая коэффициентам постоянного тока, является постоянной матрицей, так что коэффициент постоянного тока является средним (или кратным ему) пикселей в блоке размеров 8 × 8. Среднее значение пикселя в любом блоке 8×8 не будет существенно отличаться от среднего значения в соседнем блоке 8 × 8; поэтому значения коэффициентов постоянного тока будут относительно близки, эффективнее кодировать разницу между ним и значение коэффициента постоянного тока, соответствующее блоку 8x8 точек, ранее входившему в ту же категорию (Y, U или V), чем сами метки.

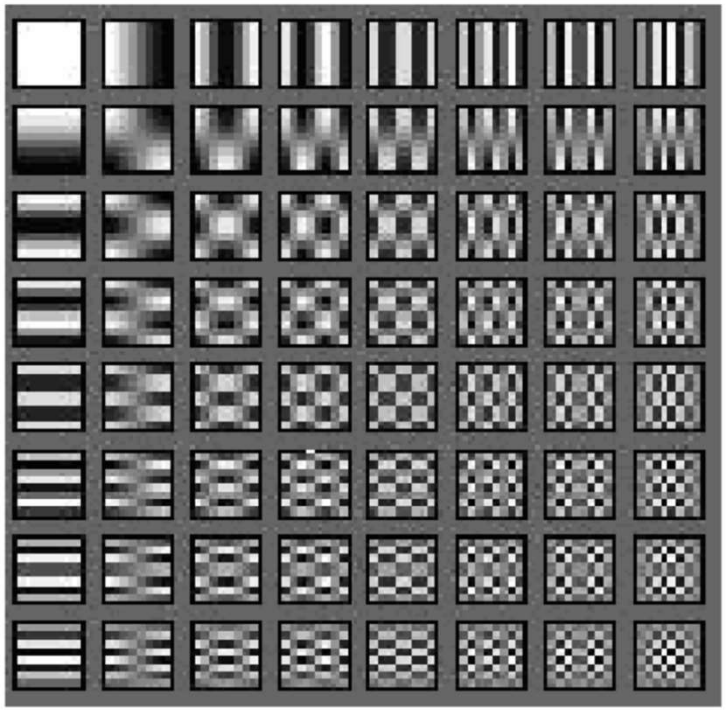


Рис. 5.13. Основные изображения для дискретного двухмерного косинусного преобразования

5.5.1. Базовый режим

В зависимости от количества битов, используемых для кодирования значения пикселя, количество значений, которые могут принимать метки, и, следовательно, различия между коэффициентами, могут стать довольно большими. Код Хаффмана для такого большого алфавита было бы трудно реализовать. Рекомендация JPEG решает эту проблему путем секционирования значений, которые могут принимать разные значения, в классы. Размер этих классов увеличивается в два в степени. Таким образом, нулевой класс имеет только один член, первый класс имеет два члена, второй класс имеет четыре члена и так далее. Номер класса кодируется алгоритмом Хаффман. Классы и соответствующие кодовые слова Хаффмана приведены в таблице 5.8. Каждая строка таблицы 5.8 содержит числа, которые больше и больше, чем в предыдущей строке, и не содержит чисел из предыдущей строки. Строка *i* содержит целые числа из диапазона [-(2*i*-1), +(2*i*-1)], из которых отсутствует средний интервал (-(2i-1-1),+ (2i-1-1)). При кодировании коэффициентов постоянного тока DC передается код, соответствующий классу *i* (строка в таблице), в который попадает число, за которым следуют биты *i*, представляющие номер столбца в таблице 5.8, в котором находится кодируемое число.

5.5.1. Базовый режим

Номер столбца, *C*, является *i*-битным представлением значения разницы, в случае положительных значений или наименее значимых *i* бит представления значения разности минус 1, в дополнении к 2, если оно отрицательное.

В случае значений в таблице 5.7, учитывая, что блок не является первым, метка, соответствующая коэффициенту постоянного тока DC, кодируется дифференциально, т.е. кодируется разница между количественным значением метки в этом блоке и количественным значением образца в предыдущем блоке. Если блок, подлежащий обработке, является первым, значение коэффициента постоянного тока DC передается четко.

Например, если значение кодируемой разницы равно 21 и принадлежит компоненту Y, сначала передается код для класса 5, который равен 111110. Значение положительное и находится в 21-м столбце строки 5 (подсчет столбцов начинается с 0), поэтому к этому коду должно быть добавлено 5-битное представление числа 21, которое является 10101, причем полный код 11111010101.

Если, однако, значение, подлежащее кодированию, равно -21, класс одинаков, разница является отрицательным значением коэффициента. Это значение находится в колонке 10 класса 5. Поэтому передается либо 5-битное представление номера столбца 10, которое равно 01010, либо передаются последние 5 бит представления в дополнение к числу 2 (-21), минус 1, то есть (-22)2C = 101010, последние 5 бит которого идентичны числу 10. Полный код в этом случае будет 11111001010.

5.5.1. Базовый режим

В случае коэффициента постоянного тока DC в таблице 5.7, если предположить, что соответствующая метка в предыдущем блоке была -1, то разница составит 3. Из таблицы 5.8 видно, что это значение относится к классу 2. Так, будет передан код Хаффмана для класса 2, 110, за которым последует последовательность из двух бит, 11, чтобы указать номер столбца, в котором находится число 3 или значение в этом классе, которое было закодировано, то есть оно передаст 11011.

Число кодовых слов Хаффмана равно логарифму, основанному на двух из числа возможных значений, которые могут принимать различия между метками. Если различия могут принимать 4096 возможных значений, длина кода Хаффмана равна $\log_2 4096 = 12$, числу, обычно используемому в кодировании.

Таблица 5.8. Различия в кодировании меток контроллеров домена

Clasa								Codul Huffman corespunzător clasei
0				0				0
1			-1		1			10
2		-3	-2		2	3		110
3	-7	.	-4		4	.	7	1110
4	-15	.	-8		8	.	15	11110
5	-31	.	-16		16	.	31	111110
6	-63	.	-32		32	.	63	1111110
7	-127	.	-64		64	.	127	11111110
8	-255	.	-128		128	.	255	111111110
9	-511	.	-256		256	.	511	1111111110
10	-1023	.	-512		512	.	1023	11111111110
11	-2047	.	-1024		1024	.	2047	111111111110
12	-4095	.	-2048		2048	.	4095	1111111111110
13	-8191	.	-4096		4096	.	8191	11111111111110
14	-16383	.	-8192		8192	.	16383	111111111111110
15	-32767	.	-16384		16384	.	32768	111111111111111
16				32768				

5.5.1. Базовый режим

Кодирование коэффициентов переменного тока АС

Порядок, в котором матрица перемещается для кодирования коэффициентов переменного тока, выбирается таким образом, чтобы наилучшим образом использовать распределение значений коэффициентов. Цель состоит в том, чтобы группировать пустые значения в строки как можно дольше, так как это позволяет более эффективно кодировать (максимальное сжатие).

Поскольку значения, отличные от нуля, сосредоточены в углу массива, построчная прокрутка неэффективна. Поэтому предпочтительна зигзагообразная прокрутка, порядок извлечения элементов из матрицы показан на рисунке 5.14.

Для матрицы коэффициентов в таблице 5.7 коэффициенты, нарисованные в следующем порядке: 1, -9, 3, 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, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0. Видно, как нулевые значения сгруппированы в конце строки. Этот факт будет учтен на следующем этапе, на котором осуществляется кодирование этих элементов. Каждый коэффициент представляет собой целое число, которое может варьировать, согласно стандарту JPEG, между -1023 и +1023 (значения получаются путем анализа наиболее неблагоприятного случая и способа работы дискретного косинусного преобразования).

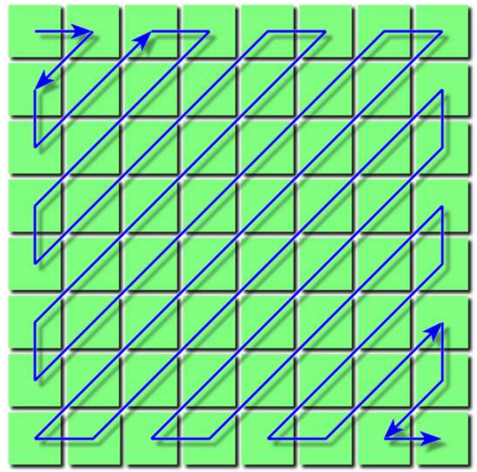


Рис. 5.14. Зигзагообразное прохождение матрицы количественных коэффициентов

5.5.1. Базовый режим

В случае «нормального» кодирования каждый из этих коэффициентов может быть представлен на 11 битах (1 бит знака + 10-битное значение). Этот режим кодирования в основном приводит к многоразрядному представлению, чем в случае несжатого изображения (для коэффициента используется 11 бит вместо 8 бит для пикселя). По этой причине в случае с алгоритмом сжатия JPEG прибегают к другому способу кодирования.

В этом случае код ассоциируется с комбинацией числа пустых значений (если таковые имеются), которые предшествуют элементу, отличному от нуля, и значения последнего. В основном пары кодируются (Количество нулей, Z – Значение, x) вместо каждого коэффициента в отдельности. В реальности, по причинам уменьшения количества возможных комбинаций, количество нулей ограничено 16. При наличии более 16 нулевых элементов выдаются специальные коды (ZRL), обозначающие 16 нулей, за которыми не следует элемент, отличный от нуля. Например, 18 нулей, за которыми следует элемент со значением -21, будут закодированы ZRL, за которым следует код, соответствующий паре (2,-21). Кроме того, если от определенной точки строки до конца строки больше нет никакого элемента, кроме нуля, вместо всех оставшихся нулевых значений выдается другой специальный код (EOB).

5.5.1. Базовый режим

В заключение, для каждого числа *x*, которому предшествуют *Z* нули, которые образуют пару (*Z, x*), энкодер должен:

– найти число последовательных нулей *Z*, предшествующих ему;

– определить строку *i* и колонку *C* таблицы 5.8, соответствующие номеру,

– идентифицировать из таблицы 5.9 по номеру *Z* и классу *i* код Хаффмана, соответствующий паре;

– слово кода, найденное Хаффманом, объединяет *i*-битное представление столбца *C*.

С учетом этих замечаний для коэффициентов, указанных в таблице 5.7, приводится следующий набор кодов, которые должны быть получены:

(0, 1)

(0, -9)

(0, -3)

(0, 0, ... 0) = EOB

Таблица 5.9. Кодирование ХАФФМАНА для коэффициентов переменного тока AC

Z/*i*	Cuv. de cod	Z/*i*	Cuv. de cod	Z/*i*	Cuv. de cod
0/0(EOB)	1010			F/0(ZRL)	11111111001
0/1	00	1/1	1100	F/1	1111111111110101
0/2	01	1/2	11011	F/2	1111111111110110
0/3	100	1/3	1111001	F/3	1111111111110111
0/4	1011	1/4	111110110	F/4	1111111111111000
0/5	11010	1/5	11111110110	F/5	1111111111111001
…					

5.5.1. Базовый режим

Например, требуется закодировать символ (0,1) в приведенной выше строке. Первое значение, 1, относится к категории 1. Поскольку нет нулей, предшествующих этому значению, передается код Хаффмана, соответствующий 0/1, который из таблицы 5.10 равен 00. Объедините с этим кодом один бит 1, чтобы указать, что передаваемое значение равно 1. Таким образом, кодовым словом для пары (0,1) является 001. Аналогичным образом, -9 является шестым элементом в классе 4. Вот почему передается двоичная строка 1011, которая является кодом Хаффмана для 0/4, за которой следует 0110, чтобы показать, что -9 является шестым элементом в классе. Следующим значением является 3, которое относится к классу 2, поэтому передается код Хаффмана 01, соответствующий 0/2, за которым следует 11. Все образцы после этой точки равны нулю, поэтому передается код Huffman EOB, который в данном случае равен 1010.

Для рассмотренного примера данные составляют 11011 001 10110110 0111 1010, то есть для представления блока размером 8x8 используется число 24 бита, то есть в среднем 3/8 бит/пиксель.

Указано, что таблицы, представленные для количественной оценки и кодирования коэффициентов постоянного и переменного тока, не являются единственными, рекомендованными стандартом JPEG, но они ограничены 4, кодовыми таблицами Хаффмана для кодирования коэффициентов переменного тока для яркости и цветности. Базовый режим JPEG использует только две такие таблицы.

5.5.1. Базовый режим

Цепочка декодирования JPEG маршрутизируется в обратном порядке кодирования. Таким образом, сжатое изображение JPEG подвергается на первом шаге энтропийному декодеру. После энтропийного декодирования применяется деквантизация, используя те же коэффициенты, которые использовались для квантования, представленные в таблицах 5.6, a, b. После деквантизации коэффициенты дискретного косинусного превращения представлены в таблице 5.10. Эти коэффициенты «отправляются» в обратный дискретный блок косинусного преобразования, IDCT, который применяет преобразование, заданное отношением 5.36. Им добавляется 128 и получают перестроенный блок, показанный в таблице 5.11. Качество этого изображения зависит от количества коэффициентов, сохраняемых при кодировании. Если сохранить все ненулевые коэффициенты, то реконструированное изображение будет очень похоже на оригинал.

Таблица 5.10. Значения коэффициентов после деквантизации

32	11	0	0	0	0	0	0
-108	0	0	0	0	0	0	0
42	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	0

Таблица 5.11. Перестроенный блок

123	122	122	121	120	120	119	119
121	121	121	120	119	118	118	118
121	121	120	119	119	118	117	117
124	124	123	122	122	121	120	120
130	130	129	129	128	128	128	127
141	141	140	140	139	138	138	137
152	152	151	151	150	149	149	148
159	159	158	157	157	156	155	155

5.5.1. Базовый режим

Хотя сокращение существенное, с 8 бит на пиксель до 3/8 бит на пиксель, воспроизведение удивительно близко к оригиналу.

Если требуется более точное воспроизведение, это может быть сделано за счет увеличения скорости, уменьшения размера шага квантования в таблице квантования. Это вызовет увеличение количества передаваемых битов. Точно так же можно снизить скорость, увеличив шаг квантования, за счет увеличения искажений.

5.5.2. Режим кодирования с расширенными потерями

Существует три основных отличия базового и расширенного режима кодирования, а именно:

— Расширенный режим может использовать 8-12 бит;

– Расширенный метод может использовать в качестве энтропийного кодирования как кодирование Хаффмана, так и арифметическое кодирование;

— Расширенный режим может использовать как прогрессивное, так и последовательное кодирование.

Во многих приложениях сжатия изображений возникает недостаток, заключающийся в том, что из-за высокого разрешения, с которым представлены изображения, время кодирования, передачи и декомпрессии, соответственно, составляет порядка минут. В таких приложениях может быть применен алгоритм прогрессивного сжатия, который изначально производит необработанное изображение, после чего при последовательных сканированиях улучшается качество изображения.

5.5.2. Режим кодирования с расширенными потерями

Если предполагается, что значение пикселя близко к значению его соседа, можно использовать значение соседнего пикселя для прогнозирования значения кодируемого пикселя. Идея состоит в том, чтобы удалить любую избыточность, которая может существовать. Разница между текущим и прогнозируемым значением кодируется и передается дальше. Эта разница называется *ошибкой прогнозирования или остатка*. Приемник использует тот же соседний пиксель для прогнозирования для этого пикселя и тот же алгоритм прогнозирования, что и передатчик. Поскольку для прогнозирования используется один и тот же пиксель и алгоритм, приемник должен генерировать то же значение прогноза, что и передатчик. Это значение, при добавлении к ошибке прогнозирования, заданной передатчиком, должно привести именно к исходному восстановленному пикселю. Если алгоритм, используемый для прогнозирования, состоит из выбора линейной комбинации соседних пикселей, то такой подход называется *линейным предсказанием*. Чтобы передатчик и приемник могли использовать один и тот же пиксель для генерации прогноза, должен быть наложен порядок пикселей. Как правило, предполагается, что пиксели изображения генерируются строка за строкой, слева направо и сверху вниз. Этот процесс называется *порядком сканирования растра*.

5.5.2. Режим кодирования с расширенными потерями

Например, считается, что «изображение» на рисунке 5.15 *a* является исходным изображением. Если левый сосед каждого пикселя используется в качестве предсказания этого пикселя, ошибка прогнозирования может быть представлена в виде остаточного изображения, как показано на рисунке 5.15, *b*.

Обратите внимание на большое количество нулей в остаточном изображении. На изображении, где в основном присутствует этот тип избыточности или структуры, то есть соседние пиксели имеют значения, близкие друг к другу, такой подход приведет к остаточному изображению, состоящему в основном из нулей и малых чисел. Остаточное изображение обычно может быть закодировано с гораздо меньшим количеством битов, чем исходное изображение. В этом примере для прогнозирования используется сосед слева. Но можно использовать и верхний сосед или выгодную их комбинацию. Схемы прогнозирования для текущего пикселя, основанные на соседях в разных направлениях, известны как двумерные схемы прогнозирования. Несмотря на свою простоту, методы линейного прогнозирования эффективны, а их производительность удивительно близка к производительности, достигнутой с помощью других, более сложных методов.

4	8	4	8	1	1	1	1
1	2	4	6	5	1	1	1
8	4	5	5	5	5	5	5
2	4	8	5	7	9	5	5
2	4	6	7	7	7	9	9
2	2	2	3	4	9	7	3
3	3	6	6	6	6	7	7
7	7	7	7	6	7	7	7

a)

4	4	-4	4	-7	0	0	0
1	1	2	2	-1	-4	0	0
8	-4	1	0	0	0	0	0
2	2	4	-3	2	2	-4	0
2	2	2	1	0	0	2	0
2	0	0	1	1	5	-2	-4
3	0	3	0	0	0	1	0
7	0	0	0	-1	1	0	0

b)

Рис. 5.15. Исходное изображение (a) и ошибка его прогнозирования (b)

5.5.2. Режим кодирования с расширенными потерями

Режим прогрессивного сжатия JPEG получается путем создания набора подизображений, и каждое подизображение будет закодировано с определенным набором коэффициентов DCT. Следовательно, DCTencoder должен иметь буфер, в котором данные (коэффициенты DCT субизображений) хранятся до энтропийного кодирования. На рисунке 5.16 показано, как формировать прогрессивно закодированные изображения JPEG.

Рис.5.16. Прогрессивное сжатие JPEG

Прогрессивное сжатие JPEG может быть получено с использованием трех типов алгоритмов:

— прогрессивный алгоритм спектрального отбора;

– прогрессивный алгоритм последовательных приближений;

– комбинированный прогрессивный алгоритм.

5.5.2. Режим кодирования с расширенными потерями

Прогрессивный алгоритм спектрального отбора группирует коэффициенты DCT в несколько частотных диапазонов. Обычно сначала передаются низкочастотные коэффициенты, после чего следуют высокочастотные коэффициенты. На рисунке 5.17 показан пример, где коэффициенты DCT разделены на четыре частотных диапазона. В полосе 1 имеются только коэффициенты постоянного тока (DC), полоса 2 содержит первые коэффициенты AC, AC1, AC2, полоса 3 содержит коэффициенты AC3, AC4, AC5, AC6 и полосу 4, коэффициенты AC7,... , AC63.

Прогрессивный алгоритм с последовательными приближениями должен основываться на начальной передаче с низкой точностью всех коэффициентов DCT, после чего остальная информация передается с целью получения высококачественного изображения. На рис. 5.18 показан пример, где коэффициенты DCT разделены на три полосы последовательных приближений: полоса 1, которая содержит все коэффициенты DCT, деленные на 4, полоса 2, содержащая коэффициенты, деленные на 2, и полоса 3, содержащая все коэффициенты DCT при номинальной точности.

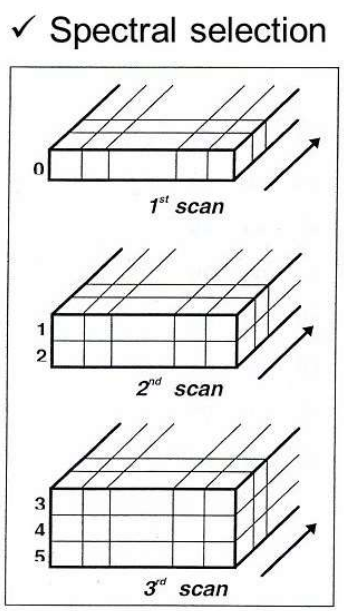


Рис. 5.17. Прогрессивный алгоритм со спектральным выделением

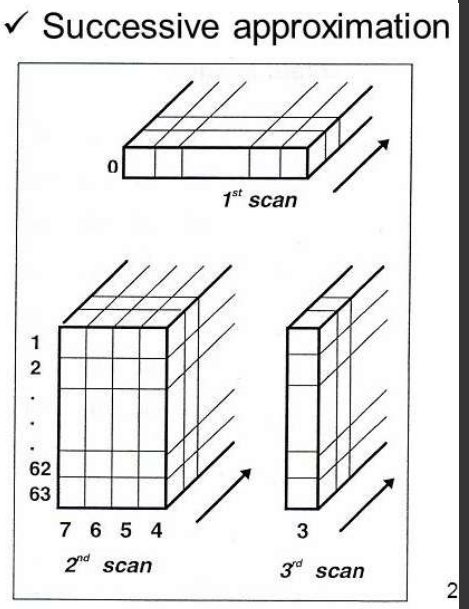


Рис. 5.18. Прогрессивный алгоритм с последовательными приближениями

5.5.2. Режим кодирования с расширенными потерями

Комбинированный прогрессивный алгоритм представляет собой комбинацию первых двух алгоритмов. На рисунке 5.19 показана плоскость коэффициентов DCT изображения после применения комбинированного алгоритма. Получено 8 последующих коэффициентов DCT, которые будут передаваться в порядке нумерации, а именно коэффициенты DC1, затем AC1 и так далее.

На рисунке 5.20 показано постепенно распаковываемое изображение JPEG для разных моментов времени.

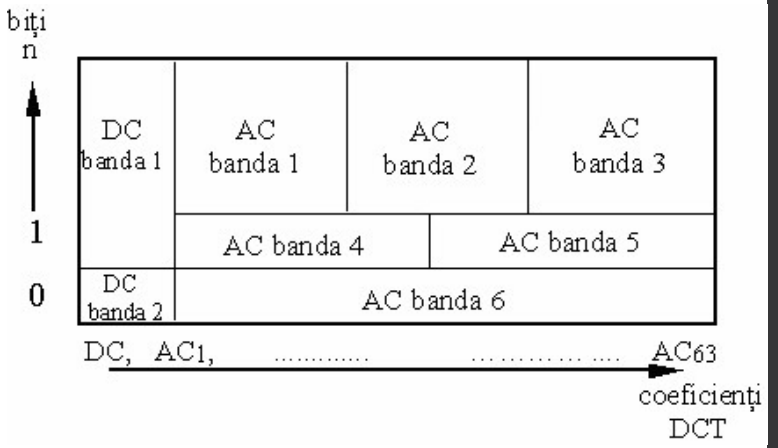
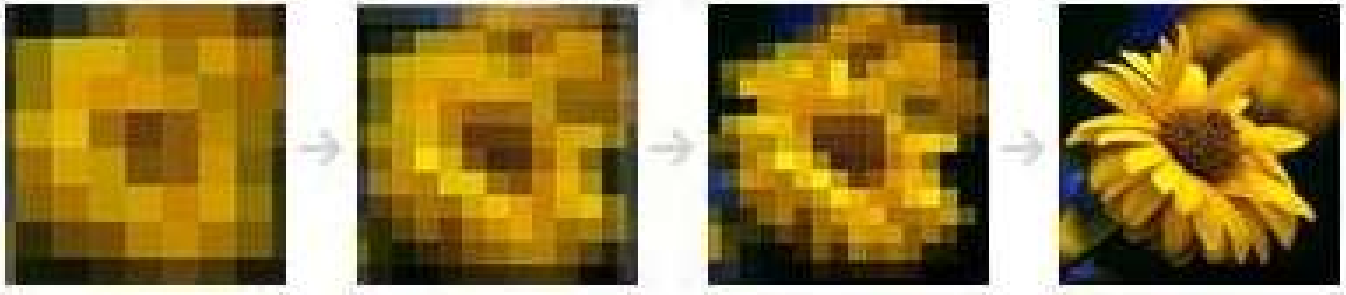


Рис.5.19 Комбинированный прогрессивный алгоритм



a) *b*) *c*) *d*)

Рис.5.20 Этапы декомпрессии изображения с использованием прогрессивного алгоритма
a) изображение в момент t_1 b) изображение в момент $t_2 > t_1$
c) изображение в момент $t_3 > t_2$ d) исходное изображение ($t = t_4$)

5.5.3. Последовательное сжатие JPEG без потерь

Стандарт JPEG также позволяет использовать алгоритм сжатия без потерь, соответственно алгоритм прогнозируемого сжатия вместо преобразования DCT. На рисунке 5.21 показана блочная схема кодировщика JPEG без потерь. Блоки неравномерного дизеринга в стандартной схеме сжатия JPEG заменяются в этом случае блоком прогностического кодирования.

Блок предиктора работает следующим образом: прогноз образца X' вычисляется на основе предыдущих выборок A, B и C, показанных на рисунке 5.22, после чего вычисляется разница между исходной выборкой X и предсказанной: DX = X – X'.

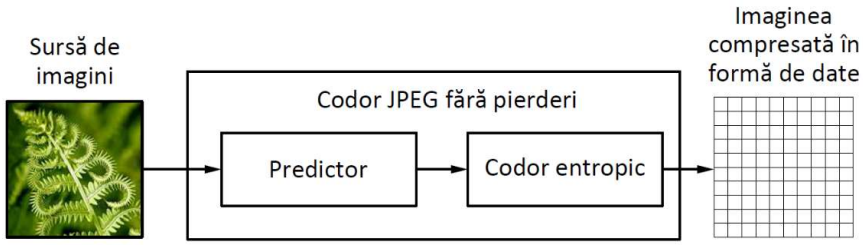


Рис. 5.21. Обобщенная схема кодировщика JPEG без потерь

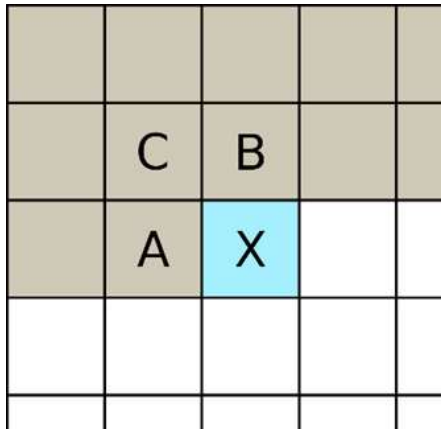


Рис. 5.22. Позиционирование выборок, на основании которых рассчитывается прогноз

5.5.3. Последовательное сжатие JPEG без потерь

Эта разница энтропически кодируется с помощью кодировщика Хаффмана. На рисунке 5.23 показаны шаги, предпринятые в алгоритме кодирования.

В таблице 5.12 приведены некоторые формулы, используемые при прогнозировании.

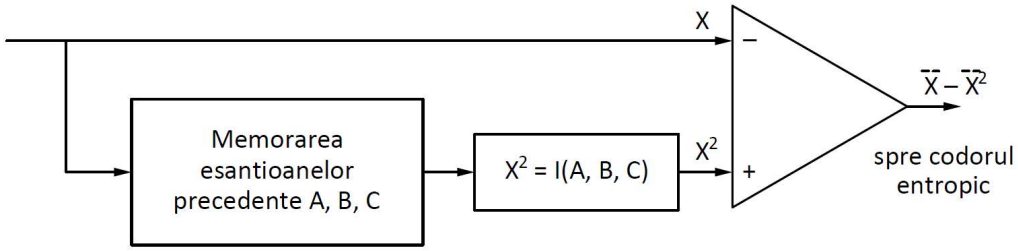


Рис. 5.23. Схема кодирования без потерь JPEG

Таблица 5.12. Предикторы, используемые для сжатия с потерями

Index	Formula de predicție
0	fără predictor
1	X=A
2	X=B
3	X=C
4	X=A+B-C
5	X=A+(B-C)/2
6	X=B+(A-C)/2
7	X=(A+B)/2

5.5.4. Иерархическое сжатие JPEG

Иерархическое сжатие JPEG позволяет прогрессивно представлять декодированное изображение аналогично прогрессивному алгоритму, но в дополнение к этому, оно позволяет закодировать изображения с несколькими разрешениями. Это создает набор сжатых изображений, начиная с изображений с низким разрешением и продолжая изображениями, разрешение которых увеличивается до разрешения исходного изображения. Этот процесс называется недостаточной дискретизацией или пирамидальным кодированием. На рисунке 5.24 показан такой алгоритм.

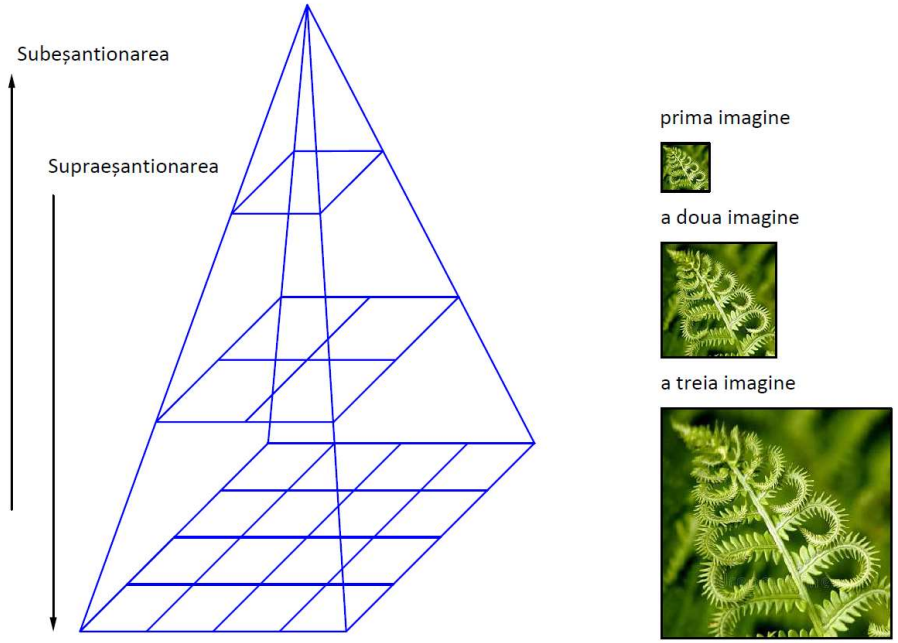


Рис. 5.24. Иерархическая кодировка JPEG

5.5.4. Иерархическое сжатие JPEG

После процесса недостаточной дискретизации каждое изображение с низким разрешением масштабируется до изображения с немедленным более высоким разрешением обратным методом, называемым перевыборкой, который используется в качестве предиктора для следующего этапа. Алгоритм иерархического сжатия состоит из следующих этапов:

— фильтрация и недостаточной дискретизация исходного изображения с нужным множителем (кратным 2), вдоль каждого измерения (по вертикали, горизонтали);

— кодирование изображения с низким разрешением с использованием методов последовательного кодирования DCT, прогрессивного DCT или кодирования без потерь;

– декодирование изображения с низким разрешением, интерполяция и сверхдискретизация в 2 раза, в горизонтальном и/или вертикальном направлении, с использованием интерполяционного фильтра, идентичного фильтру декодера;

— использование чрезмерно выборочного изображения в качестве предиктора исходного изображения и кодирование различий между двумя изображениями, используя один из упомянутых методов;

– повторять алгоритм до тех пор, пока изображение не будет закодировано в исходном разрешении.

Иерархическое кодирование требует достаточно большой области памяти, чтобы иметь возможность реализовать алгоритм, но преимущества заключаются в том, что закодированное изображение сразу же доступно в разных разрешениях.

https://www.youtube.com/watch?v=CJFUN6BrkGE

КАК РАБОТАЕТ СЖАТИЕ?

ВОПРОСЫ