

ЛР5: Сжатие изображений.

Цель: Сжатия изображения на примере стандарта JPEG.

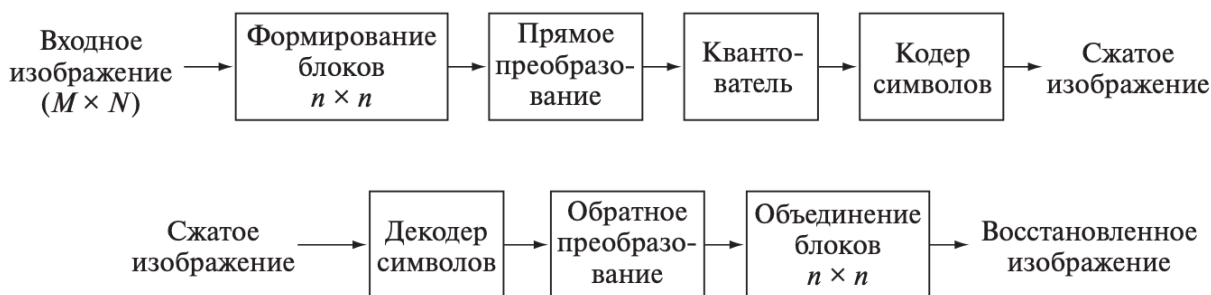
Теоретическая часть

Блочное трансформационное кодирование

Блочное трансформационное кодирование используется в следующих стандартах сжатия и файловых форматах: JPEG, М-JPEG, MPEG-1,2,4, H.264 и т.д. В данной работе будет рассмотрен метод сжатия изображений, используемый в стандарте JPEG. Метод сжатия основан на разбиении изображения на небольшие неперекрывающиеся блоки одинаковых размеров (8×8) и последующей независимой обработке блоков при помощи двумерных преобразований (ДКП).

В блочном трансформационном кодировании обратимое линейное преобразование (ДКП) используется для отображения каждого блока в набор коэффициентов преобразования, которые затем квантуются и кодируются. Для большинства реальных изображений значительное число коэффициентов имеют малую величину и могут быть достаточно грубо квантованы (или полностью удалены) ценой небольшого искажения изображения.

На рисунке схематически показана типичная система блочного трансформационного кодирования. Кодер выполняет четыре достаточно понятные операции: разбиение изображения на блоки, преобразование, квантование и кодирование. Декодер выполняет обратную последовательность операций (за исключением квантования). Первоначально изображение размерами $M \times N$ разбивается на MN/n^2 блоков размерами $n \times n$, которые затем и подвергаются преобразованиям.



Целью процесса преобразования является декорреляция значений элементов в каждом блоке или уплотнение как можно большего количества информации в наименьшее число коэффициентов преобразования. В качестве декоррелирующего преобразования выступает ДКП формата 8. В результате преобразования получим разделенное на 64 канала (8×8) изображение.

Квантование

На этапе квантования грубо квантуются или же удаляются те коэффициенты, которые несут малое количество информации в заранее заданном смысле (ВЧ часть изображения). Эти коэффициенты дают наименьший вклад в качество восстанавливаемого блока. Таким образом задачу квантования можно сформулировать следующим образом: распределить заданное число бит b_i по каналам при условии минимизации ошибки реконструкции сигнала. Условие оптимального распределения бит выполняется, если коэффициенты b_i рассчитать в соответствии со следующим выражением:

$$b_i = b + \frac{1}{2} \log_2 \frac{\epsilon \sigma_k^2}{\prod_{k=0}^{63} [(\epsilon \sigma_k^2)^{1/64}]} \quad (1.1)$$

где ϵ — константа равная $1/12$; σ_k^2 — дисперсия сигнала в k -м канале; b — заданное число бит на изображение. Для тестового изображения *lena_gray_256.tif* и числа бит на изображение $b = 4$ коэффициенты b_i равны

	1	2	3	4	5	6	7	8
1	8.7688	7.1776	6.1494	5.3754	4.7440	4.4447	4.0693	3.7315
2	6.3118	6.0979	5.4654	4.8793	4.4866	4.1675	3.8472	3.7521
3	5.1980	5.1275	4.9493	4.7050	4.2024	3.9913	3.6881	3.4014
4	4.2074	4.4341	4.5504	4.2855	4.1849	3.8022	3.7622	3.4498
5	3.7554	3.9138	3.9468	3.9639	3.8045	3.8026	3.4960	3.4499
6	3.2701	3.3409	3.3985	3.5131	3.4283	3.3764	3.2540	3.0983
7	2.9715	3.1137	3.1357	3.0489	3.1945	3.1088	3.0639	3.0899
8	2.7423	2.8166	2.8214	2.8371	2.8962	3.0349	2.9693	2.9347

Используя коэффициенты b_i выполняется квантование 64 канальных сигналов:

$$q = \text{round}(p h) / h \quad (1.2)$$

где p — входное значение; q — выходное значение.

Кодирование символов

На заключительном этапе работы кодера изображений осуществляется кодирование квантованных коэффициентов, которое состоит из следующих шагов:

1. Квантованные коэффициенты <<вытягиваются>> в вектор, путем зигзагообразного сканирования.

[illegible]

2. Конец вектора соответствует ВЧ каналам изображения, которые квантуются грубо и следовательно будут равны нулю. Сжатие последовательностей, состоящей из одинаковых символов выполняется с помощью *кодирования длин серий* (run-length encoding, RLE). Алгоритм заменяет повторяющиеся символы на один символ и число его повторов. Например строка состоящая из 51 символа

WWWWWWWWBBBWWWWWWWWWWWWWWWWWWWWWWWWWWWWWBWWWWWWWWWWWWWW

С помощью алгоритма RLE запишется следующим образом

9W3B24W1B14W

Рекомендуется хранить символы и число их повторений в разных массивах (каждый символ занимает 1 байт, - показан для наглядности)

W-B-W-B-W
9-3-24-1-14

3. На заключительном шаге выполняется статистическое кодирование всех каналов (RLE-символов и RLE-повторов) отдельно с помощью алгоритма Хаффмана.

Ход работы

1. Квантование коэффициентов ДКП:

- a) Загрузить изображение из папки *test_images* в соответствии с вариантом с помощью команды *imread*. Преобразовать в полутоновое изображение и обрезать до размеров 256x256.
- b) Выполнить анализ изображения с помощью ДКП из ЛР5 и получить 64 канальных сигнала.
- c) Рассчитать дисперсию и рассчитать бюджет бит b_i по формуле (1.1) в каждом канале (команды *var*, *prod*, *log2*). Рекомендуется отладить шаг на изображении *lena_gray_256.tif*, значения смотреть в теоретической части.
- d) Округлить b_i с помощью команды *round*.
- e) Приравнять к 0 значения b_i меньше 0.
- f) Выполнить квантование канальных сигналов.
- g) Выполнить синтез квантованного изображения с помощью ДКП из ЛР5.
- h) Оценить качество квантованного изображения с помощью команды *psnr*.

2. Кодирование коэффициентов ДКП:

- a) Преобразовать квантованный канальный сигнал в вектор с помощью функции *zigzag*.
- b) Написать собственную функцию RLE. Символы и число их повторений хранить в разных массивах.
- c) Выполнить кодирование всех каналов (RLE-символов и RLE-повторов) отдельно с помощью алгоритма Хаффмана (команды *huffmandict*, *huffmanenco*).
- d) Рассчитать минимально требуемое число бит информации B для восстановления изображения (код и словарь Хаффмана).

3. Выполнить шаги 1 и 2 для $b = 1, 2, 3, 4, 5, 6, 7, 8$ и построить графики зависимости *psnr* от B .