Алимагадов Курбан, ПМ-11М

Отчётное домашнее задание №4

Задание 1

Перед выполнением задания изучите MATLAB-реализацию адаптивного арифметического кодера (см. файлы arencode.m и ardecode.m, а также используемые ими файлы-скрипты), сравнив её с использовавшейся ранее реализацией на языке C (см. ar0.cpp). Сравните характеристики двух реализаций по времени выполнения и эффективности сжатия тестовых данных. Какие наблюдаются различия и чем они обусловлены?

Размер файла до сжатия: 9 348 байт.

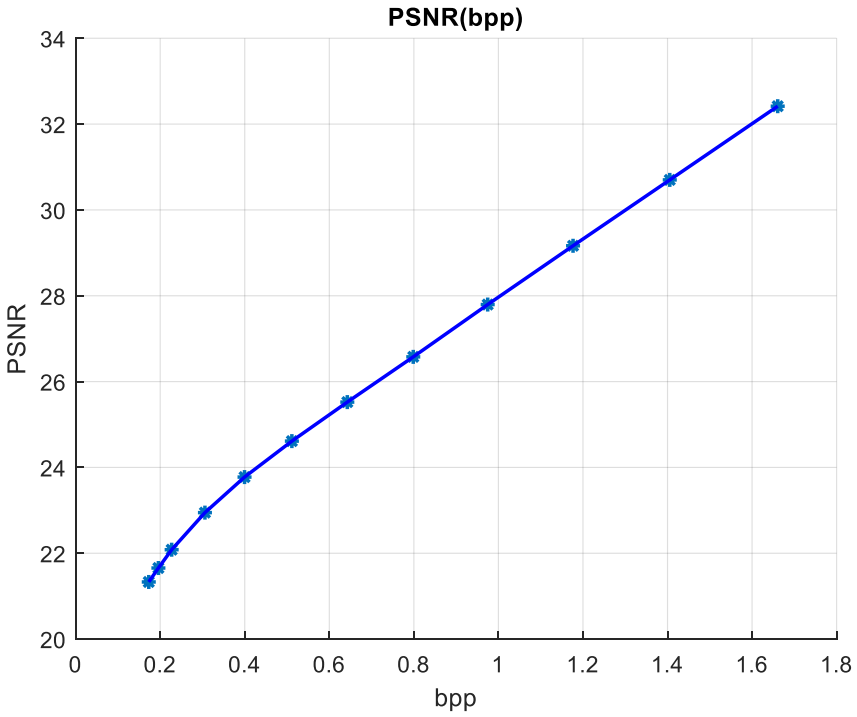
Полученные результаты:

Реализация	Размер после сжатия	Время кодирования	Размер после декодирования	Время декодирования
MATLAB	6 112 байт	2.8648 секунд	9 348 байт	3.3849 секунд
C	6 424 байт	0.012 секунд	9 348 байт	0.062 секунд

Обе реализации показывают примерно одинаковый результат эффективности сжатия, однако скорость выполнения кодирования и декодирования у реализации на C больше, это связано с тем, что реализация на C – скомпилированная программа, которая выполняется быстрее, чем интерпретация кода скрипта MATLAB.

1. Для «базового» кодера, который реализуется скриптом jp.m, постройте график зависимости PSNR(bpp) для изображения, соответствующего вашему варианту. Для этого потребуется варьировать параметр quality, который выбирается из диапазона целых чисел [1..100].

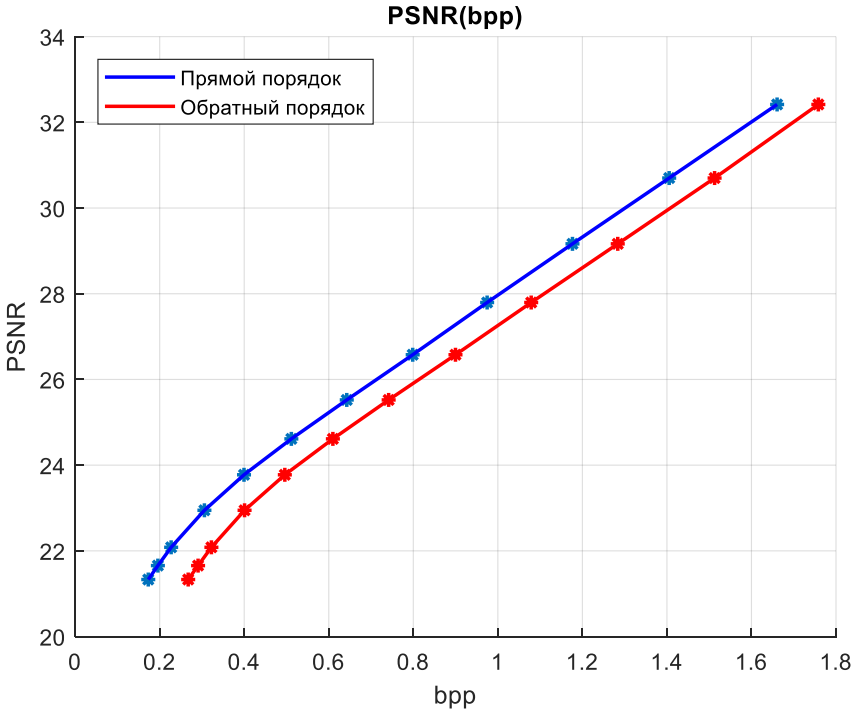
Рассмотрим график PSNR(bpp), где bpp зависит от параметра quality, который принимает значения [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100].



Видно, что с увеличением количества бит на пиксель, растёт значение PSNR.

2. В реализации jp.m используется арифметический кодер с одной адаптивной статистической моделью, последовательно применяющейся к проквантованным коэффициентам ДКП, начиная с НЧ-компонент и заканчивая ВЧ-компонентами (порядок обработки определяется «змейкой», последовательность перебора индексов ДКП задаётся 64-элементными массивами ii и jj). Изменив порядок просмотра «змейки» на обратный (от ВЧ к НЧ), проанализируйте изменение характеристик PSNR(bpp). Дайте объяснение полученного результата.

Рассмотрим графики PSNR(bpp), где bpp зависит от параметра quality, который принимает значения [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100].



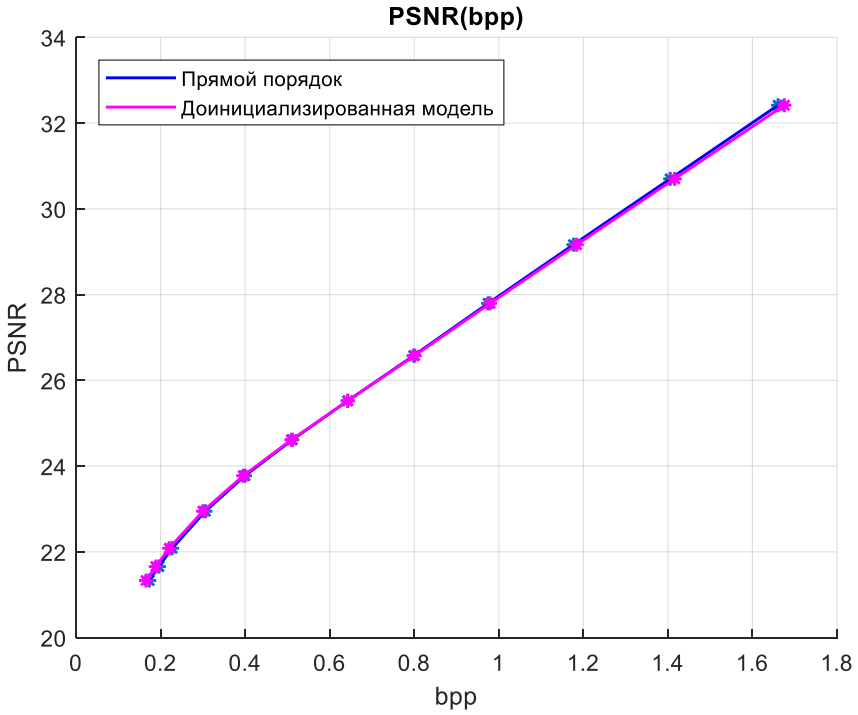
При обратном порядке прохода при тех же значениях PSNR (как и при прямом порядке обхода) наблюдается увеличение значений bbp.

3. Включите «доинициализацию» модели арифметического кодера, активировав исходно закомментированную инструкцию (в 2 местах в файле jp.m):

```
cum_freq(130:NO_OF_SYMBOLS+1) = cum_freq(130:NO_OF_SYMBOLS+1) + 6000;
```

Дайте объяснение наблюдаемым изменениям характеристик сжатия.

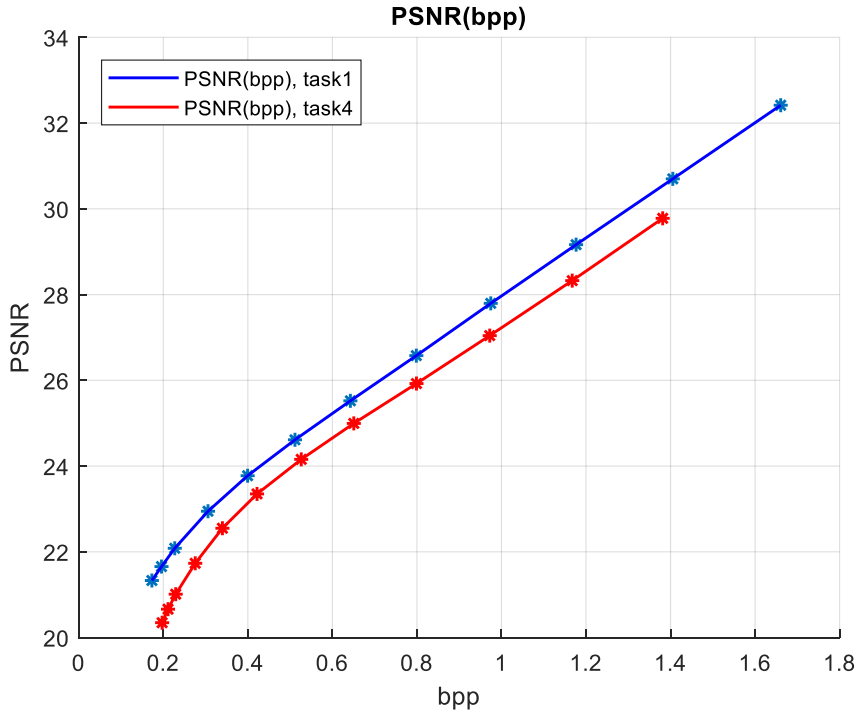
Графики построенные по полученным данным:



Результаты сжатия практически не изменились, однако по графику можно видеть, что при неизменных значениях PSNR для случаев, когда сжатие производилось с маленькими значениями quality, величина bpp немного уменьшилась (первые 6 точек фиолетового графика расположены левее соответствующих точек синего графика), а для остальных случаев величина bpp немного увеличилась (последние 6 точек фиолетового графика расположены правее соответствующих точек синего графика).

4. Сохраняя изменения алгоритма, примененные в пп. 2 и 3, замените квантование коэффициентов ДКП с округлением с шагом q на квантование с мёртвой зоной шириной 2q (для этого измените функции quantize.m и dequantize.m). Постройте график PSNR(bpp) для алгоритма сжатия, полученного в результате модификаций в пунктах задания 2–4, и сравните с графиком, который был построен в пункте 1. Сделайте выводы.

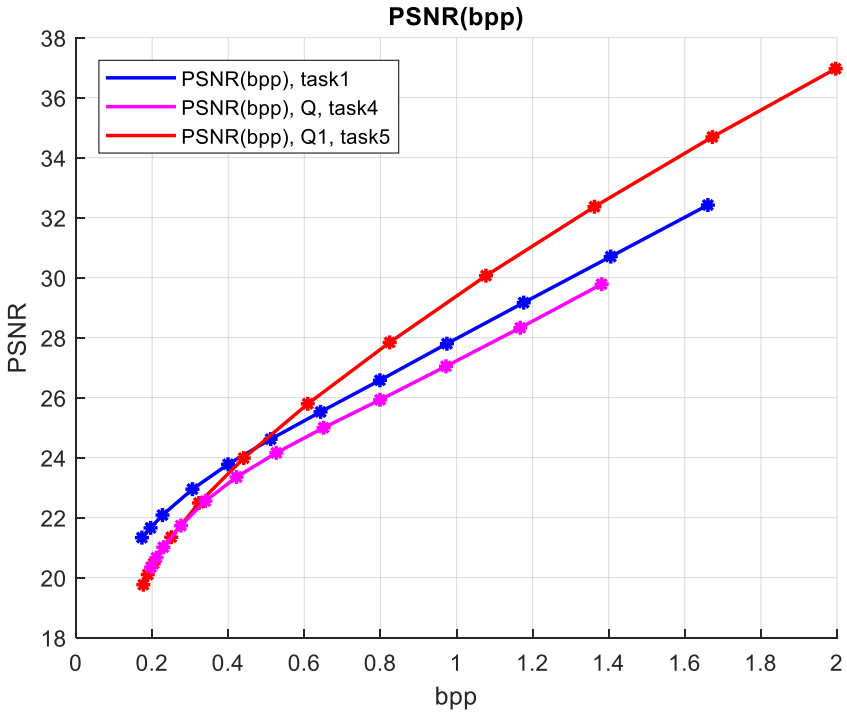
Рассмотрим графики PSNR(bpp), где bpp зависит от параметра quality, который принимает значения [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100].



По графикам видно, что изменения, сделанные в пунктах 2-3, в сочетании с квантованием с мёртвой зоной приводят к уменьшению PSNR, при тех же битовых затратах.

5. Для алгоритма сжатия, полученного в результате применения модификаций в пп. 2–4, замените матрицу квантования Q (матрица Пеннебакера) на матрицу квантования Q1 из одинаковых элементов. Постройте соответствующий график PSNR(bpp) для алгоритма сжатия. Сравните визуальное качество изображений и величину PSNR, наблюдаемых при одном и том же значении bpp с разными матрицами квантования (Q и Q1). Сделайте выводы.

Рассмотрим графики PSNR(bpp), где bpp зависит от параметра quality, который принимает значения [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100].



Для рассматриваемого изображения при использовании матрицы квантования Q1 и битовых затратах больших 0.32 бита (quality $\geq$ 30) наблюдаются большие значения PSNR, по сравнению с результатами кодера из задания 4, однако визуальное качество изображений, обработанных кодером из задания 5, заметно хуже.

Изображения, полученные в заданиях 4 и 5 для различных значений параметра quality {30, 50, 100} приведены ниже.

Матрица квантования Q (quality = 30)

Матрица квантования Q1 (quality = 30)

Матрица квантования Q (quality = 50)

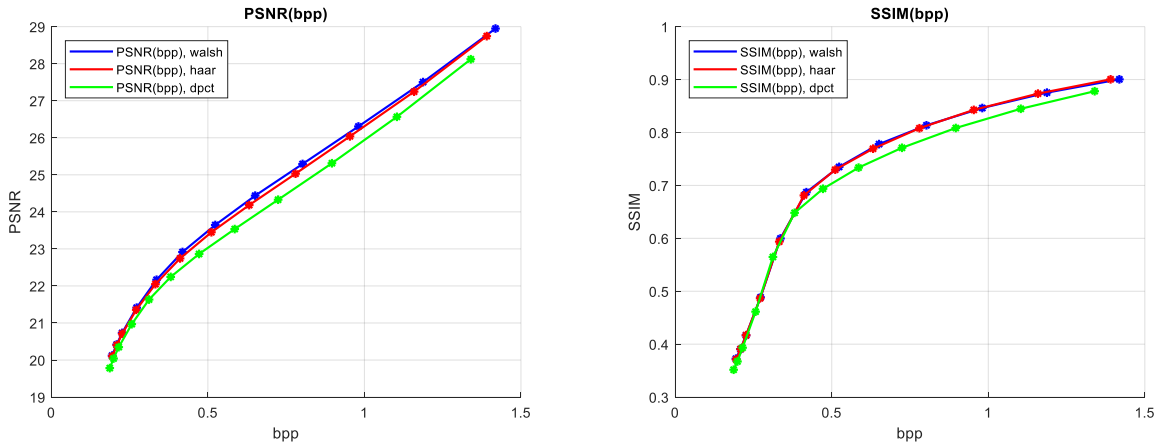
Матрица квантования Q1 (quality = 50)

Матрица квантования Q (quality = 100)

Матрица квантования Q1 (quality = 100)

6. Для итогового алгоритма сжатия, полученного в результате выполнения пп. 2–4, проведите исследования характеристик сжатия, которые получаются при использовании альтернативных дискретных ортогональных преобразований размерности 8x8: Уолша, Хаара, дискретного псевдокосинусного (см. [1], конспекты и презентации аудиторных занятий). Для этого постройте соответствующие каждому преобразованию графики PSNR(bpp) и SIMM(bpp), которые дал рассматриваемый алгоритм сжатия изображений. Сделайте выводы. Согласуются ли наблюдаемые результаты с теоретическими оценками, вытекающими из анализа средней избыточной энтропии [1] для модели марковского процесса первого порядка?

Рассмотрим графики PSNR(bpp) и SSIM(bpp), где bpp зависит от параметра quality, который принимает значения [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100].



По графикам видно, что при сжатии наименьшие битовые затраты (бит на пиксель) наблюдаются для дискретного псевдокосинусного (зелёные графики) при примерно одинаковых значениях PSNR(bpp) и SSIM(bpp), что соответствует теоретическим сведениям. При значениях bpp > 0.22 (quality > 10) PSNR и SSIM принимают наибольшие значения для преобразования Уолша (синние графики), за исключением точки, где quality = 100. В этой точке $SSIM_{Haar}$ = 0.9005, $SSIM_{Walsh}$ = 0.9004, однако разница незначительна.

Примеры изображений для значения quality = 30 приведены ниже.

Преобразование Уолша

Преобразование Хаара

ДПКП

Примеры изображений для значения quality = 50 приведены ниже.

Преобразование Уолша

Преобразование Хаара

ДПКП

Примеры изображений для значения quality = 90 приведены ниже.

Преобразование Уолша

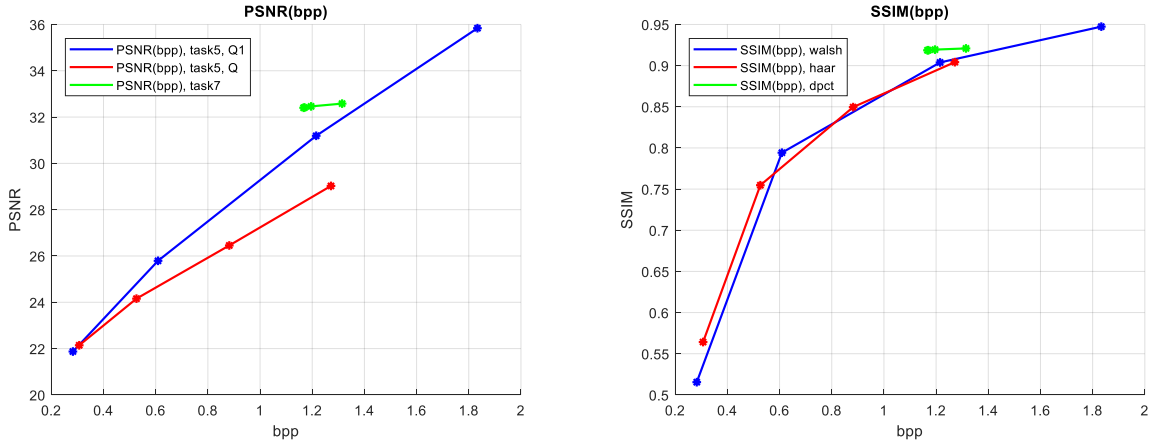
Преобразование Хаара

ДПКП

7. Ознакомившись со скриптом простейшего вейвлет-кодека jp2.m (и вызываемых им процедур encode_subband, decode_subband) по результатам экспериментов постройте зависимости PSNR(bpp) для изображения вашего варианта, выбирая количество уровней вейвлет-разложения Levels = 1, 2, 3, 4, 5. Баланс между величиной сжатия и внесённой при кодировании ошибкой устанавливается изменением параметра qstep. Сравните характеристики реализованного алгоритма вейвлет-сжатия и алгоритма на основе ДКП,

который исследовался в п.5 задания (использовать скалярное квантование с мёртвой зоной). Сделайте выводы.

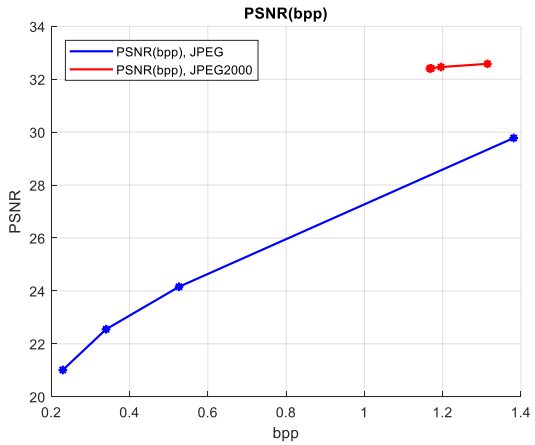
Рассмотрим графики PSNR и SSIM, где для алгоритма из задания 5 quality $\in \{25, 50, 75, 90\}$, и для алгоритма из задания 7 Levels $\in \{2, 3, 4, 5\}$.

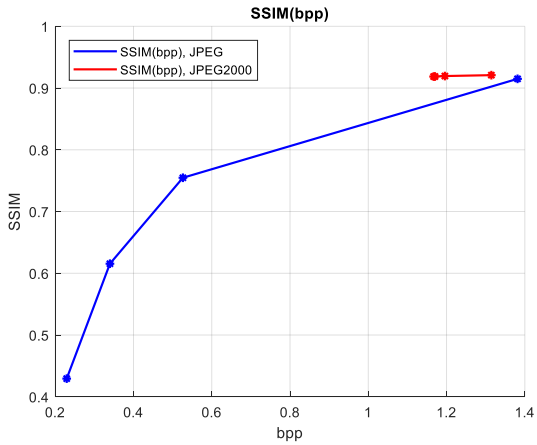


Видно, что для значений Levels $\in \{2, 3, 4, 5\}$ (и bpp ≤ 1.3) вейвлет-кодек jp2.m демонстрирует значения PSNR и SSIM большие, чем у алгоритма из задания 5, однако битовые затраты при кодировании кодеком из 7 задания не опускаются 1.16 бит на пиксель. Также можно заметить, что для значения quality = 95 (bpp = 1.8) алгоритм из 5 задания демонстрирует большие значения PSNR и SSIM, чем у вейвлет-кодека jp2.m при 5 уровнях вейвлет-разложения.

8. Исследуйте характеристики сжатия стандартных методов JPEG и JPEG 2000 с тестовым изображением вашего варианта, используя любое программное средство, которое поддерживает сохранение фотографических изображений в данных форматах (например, MATLAB: см. встроенную справку по процедуре imwrite). Для этого постройте на одном рисунке два графика PSNR(bpp), а на другом – два графика SSIM(bpp), соответствующих данным стандартным методам. Сравните с характеристиками сжатия, достигнутыми при выполнении пп. 5-7, и сделайте выводы.

Рассмотрим графики PSNR и SSIM, где для алгоритма JPEG quality $\in \{10, 30, 50, 100\}$, и для алгоритма JPEG2000 Levels $\in \{2, 3, 4, 5\}$.





По графикам можно видеть, что алгоритм JPEG2000 позволяет добиться больших значений PSNR и SSIM для любых значений параметров quality и Levels, однако при его использовании для битовых затрат мы получаем значения не меньше 1.16 бит на пиксель.

Выводы:

В работе был рассмотрен алгоритм сжатия JPEG, его модификации, основанные на замене ДКП на ДПКП, ДПУ, ДПХ, а также алгоритм JPEG2000. Установлено, что при кодировании с использованием ДПКП удаётся получить наименьшие битовые затраты при примерно одинаковых значениях метрик PSNR и SSIM, по сравнению с модификациями с использованием ДПУ и ДПХ, что соответствует теории. При сравнении алгоритмов сжатия JPEG и JPEG2000 установлено, что второй позволяет добиться больших значений метрик PSNR и SSIM при любых значениях варьируемых параметров.

Код на языке MATLAB в файлах с названиями task_i.m, где i – номер задания.