

УДК 004.421

Алгоритм сжатия изображений на основе динамического сжатия палитры

Амиантов Н.И., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»*

*Научный руководитель: Рудаков И.В., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
irudakov@bmstu.ru*

Алгоритмы сжатия изображений в той или иной форме, с потерями качества и без, и с разными областями применения используются для уменьшения занимаемого изображением пространства уже достаточно давно. Существует несколько известных подходов к сжатию изображений, алгоритмов и вспомогательных шагов [1]. Мы представляем новый метод сжатия изображений, основанный на сжатии палитры, обладающий полезным набором свойств.

Считается, что исходные изображения представлены в несжатом растровом виде, т.е. как массивы байт, обозначающих интенсивность каждого канала изображения. Исходные изображения могут быть представлены одним каналом яркости (изображения в оттенках серого) или как сумма трёх каналов красного, зелёного и синего цветов. Изображение после расшифровки должно быть представлено в схожем формате. Подразумевается, что алгоритм сжатия не используется для сжатия видео, а значит не создан для сжатия и распаковки в реальном времени, так что в процессе работы программы могут быть использованы более ресурсоёмкие, но дающие лучший результат алгоритмы.

- Формат будет использовать сжатие с потерями данных;
- Формат будет использоваться для сжатия статических изображений;
- Формат будет разрабатываться для сжатия фотографических изображений, но с сохранением границ, позволяющем использовать его для сканированных книг и текстов.

Предложенный нами формат использует идею, которая не применялась до сих пор в популярных форматах сжатия изображений – динамический выбор палитры и

степени сжатия для каждого блока по отдельности, и разбиение блоков на подблоки со своей палитрой. Похожая, но менее полная технология используется в PNG [2]. Мы собираемся задать формат файла, а также описать алгоритм сжатия без указания конкретных способов выполнения некоторых этапов (которые могут быть выбраны несколькими способами и изучались в исследовательской части).

Заголовок формата содержит информацию о размере изображения и список каналов, о которых будет сказано ниже. Структура формата в бинарном виде представлена на рисунке 1. Также в заголовке присутствует общее количество каналов.



Рис. 1. Заголовок формата

Дальнейшие структуры формата будут рассматриваться одновременно с алгоритмами, используемыми для кодирования изображения.

На первых этапах изображение должно быть преобразовано к палитре $Y C_b C_r$. Матрицы преобразований из RGB в эту цветовую палитру и наоборот есть в открытом доступе, например, в 3. Это позволяет использовать более сильное сжатие для цветowych каналов, используя особенности человеческого зрения [4].

Для каждого канала изображение кодируется отдельно, со своими настройками сжатия. Это позволяет детально настраивать процесс кодирования, ухудшая сжатие цветовой составляющей и подстраивая настройки под разные типы изображения. Заголовок канала состоит из его типа (составляющей, которую он представляет) в строковом виде, например, “Cb”, списка типов блоков, которые могут содержаться в этом канале, и списка блоков изображения. В бинарном виде заголовки каналов изображения построены как показано на рис. 2. Количество типов блоков записывается в бинарном виде отдельно, однако количество самих блоков не записывается – оно вычисляемо из информации о типах блоков и размере изображения.



Рис. 2. Заголовок канала

Для изображения задаётся несколько возможных режимов кодирования блоков.

Основным свойством блока является его размер – в частности, размер наибольшего блока будет использоваться для всех блоков первого уровня в изображении. Изображение разбивается согласно этому размеру и каждый блок обрабатывается отдельно. Также для каждого блока содержится битрейт (количество бит, затрачиваемое на каждую полную интенсивность), размер пикселя (при применении масштабирования) и размер палитры для данного режима кодирования. Для данных блоков и ниже применяется битовое сжатие – данные записываются без выравнивания по байтам. Из-за особенностей кодирования это сильно влияет на размер изображения. Важным параметром канала также является общее число возможных режимов кодирования, которое влияет на длину поля типа блока у каждого блока канала. Бинарный формат информации о блоке представлен на рисунке 3.



Рис. 3. Информация о режиме кодирования

Формат использует два уровня блоков. Размер блоков верхнего уровня равен наибольшему размеру блока в канале (как сказано выше). Все блоки второго уровня (подблоки) внутри блока сжаты одинаковым способом. В таком блоке содержится индекс режима кодирования, который используется для всех подблоков. В бинарном представлении индекс занимает минимальное количество места, исходя из количества возможных режимов кодирования для данного канала. Затем такой блок хранит трёхмерный массив интенсивностей и массив индексов, о которых будет рассказано ниже. Бинарное представление блока верхнего уровня представлено на рисунке 4.

При использовании уменьшения масштаба (пиксели размером больше 1x1) перед обработкой матрица блока разбивается на блоки размером с будущие пиксели, и формируется новая матрица, значения в которой равны средним значениям блоков исходной.



Рис. 4. Блок верхнего уровня

Блоки верхнего уровня делятся на подблоки согласно выбранному режиму

кодирования. Для каждого подблока имеется конечная последовательность в трёхмерном массив интенсивностей, значения которой соответствуют i -ым интенсивностям индивидуальной палитры подблока. Поскольку все подблоки сжимаются с одним режимом кодирования, длины палитр для всех подблоков совпадают. Все точки подблока сводятся к одному из значений индивидуальной палитры. Индексы палитры хранятся побитово в зависимости от размера палитры, и подблоки хранятся подряд, что обеспечивает высокую плотность информации.

Каналы целиком сжимаются алгоритмом LZMA2, который обеспечивает дополнительное сжатие. Такая обработка обеспечивает дальнейшее уменьшение объёма до 30% от исходной уже сжатой картинки [5].

Раскодирование такого формата – достаточно быстрая операция, не требующая больших объёмов оперативной памяти или сложных вычислений. Основная вычислительная операция – распаковка сжатых алгоритмом LZMA2 каналов. После распаковки достаточно считывать блоки подряд и заполнять матрицу будущих интенсивностей поблочно. Размеры блоков известны до их считывания. Во время чтения подблоков значения индексов подменяются в конечном раскодированном изображении на значения из палитры. Полная схема раскодирования формата представлена на рисунке 5.



Рис. 5. Схема алгоритма распаковки

Мы задали предложенный формат хранения изображений с потерей качества. Полная схема алгоритма сжатия представлена на рисунке 6. Данный формат

однозначно раскодируется, но сильно зависит от методов сжатия, которые будут описаны ниже. Плюсами формата, наблюдаемыми на этом этапе, являются:

- Простота раскодирования;
- Возможность параллельного сжатия изображения;
- Компактность при условии верного выбора палитр и режимов сжатия;
- Большие возможности по модификации изображений без перепакетки и дальнейшей потери данных.

При условии верного выбора палитр формат должен показывать хорошее сжатие и качество на различных графиках, диаграммах, элементах интерфейса и других изображениях с фиксированными палитрами и/или чёткими границами. Также такой формат должен показывать по меньшей мере приемлемое сжатие для фотографических изображений при полном сохранении резких границ и переходов, что способствует визуально лучшему качеству.

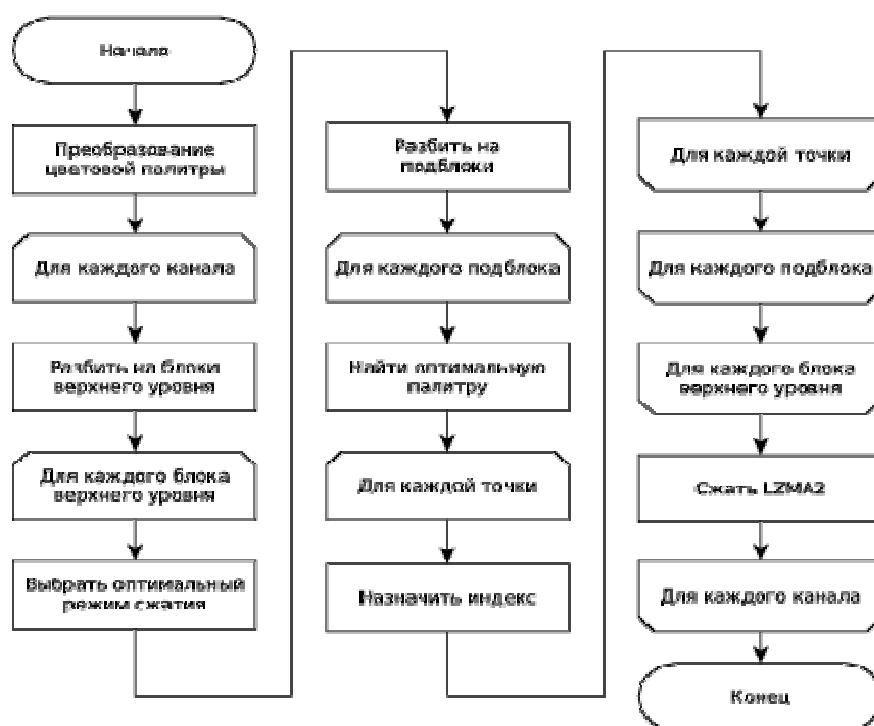


Рис. 6. Общая схема алгоритма сжатия

Для определения оптимального сжатия блоков второго уровня используется предложенная нами метрика ошибки кодирования. Предположим, нам даны два изображения (представленные в виде матриц) A и B . Наша задача – получить число e , достоверно описывающую степень отклонения изображений друг от друга. Суть алгоритма в следующем:

Вычтем изображения друг из друга. Для определённости пусть $C = A - B$. Разобьём матрицу C на матрицы размером 3×3 или меньше (на границах изображений). В дальнейшем будем рассматривать эти матрицы по одной.

Наша задача – для каждой такой матрицы найти наибольший связанный участок, все элементы которого имеют один знак. Такая область называется “серией”. Мы выбираем области одного знака, поскольку такая область хорошо отражает однородный пространственный всплеск ошибки в изображении. Обычно это труднорешаемая задача, требующая аналогов алгоритмов на графах для поиска максимального связанного подграфа. Однако, для блоков 3×3 есть две особенности, позволяющие сильно ускорить этот процесс.

Все точки блока 3×3 лежат рядом с центральной точкой блока, так что взяв серию, состоящую из центральной точки блока и других его точек одного с ней знака, можно быстро получить, вероятно, достаточно большую серию. Исключение составляет случай, когда центральная точка в матрице равна нулю – тогда любая ненулевая точка кроме неё может считаться с ней в одной серии, так что для нахождения серии достаточно найти первую такую точку.

После нахождения серии мы действуем по одному из двух способов:

- Находим сумму квадратов точек серии: $\sum x^2$;
- Находим сумму модулей точек серии: $\sum |x|$.

Полученное число назовём S . В итоге искомой метрикой для блока будет число

$$e = \frac{S}{N}$$

, где N – количество точек в серии, а общей метрикой – кортеж e_i для всех блоков.

Эту метрику мы применяем следующим образом. Сначала для нашего лучшего варианта сжатия мы считаем изменённую метрику e_{bi} :

$$e_{bi} = k \cdot \min(e_i, t)$$

, где k – коэффициент допустимости ошибок, а $t > 0$ – отсекающий фактор. Далее будет пояснена его необходимость.

Далее мы будем считать ошибку сжатия для каждого из проверяемых уровней сжатия, начиная с наихудшего. Пусть последовательность ошибок для очередного уровня – e_{ti} . Длины последовательностей e_{ti} и e_{bi} совпадают, сравним попарно соответствующие ошибки. Если хоть одна ошибка из e_{ti} меньше чем текущая, текущий уровень сжатия отбрасывается и проверяется следующий.

Таким образом, коэффициент k контролирует допустимую расходимость ошибок между лучшим вариантом сжатия и текущим. Отсекающий фактор t необходим в случае, если какие-либо блоки целиком описываются заданными палитрами (ошибка равна нулю). В таком случае один такой блок отбраковывает все уровни сжатия, которые неидеально сохраняют это состояние. Фактор t , который выбирается достаточно небольшим, препятствует такому раскладу.

Данная метрика позволяет с достаточно хорошими результатами выбирать подходящие уровни сжатия. Использование разных способов нахождения s даёт разные результаты. Так, сумма квадратов сильнее выделяет резкие пики. Пример сравнения статистики блоков для двух вариантов s представлен в таблице 1. Схема алгоритма выбора оптимального уровня сжатия блока представлена на рисунке 7.

Таблица 1

Статистика для разных вариантов s

Уровень	χ^2	$ x $
1	18%	20%
2	8%	12%
3	12%	14%
4	62%	54%

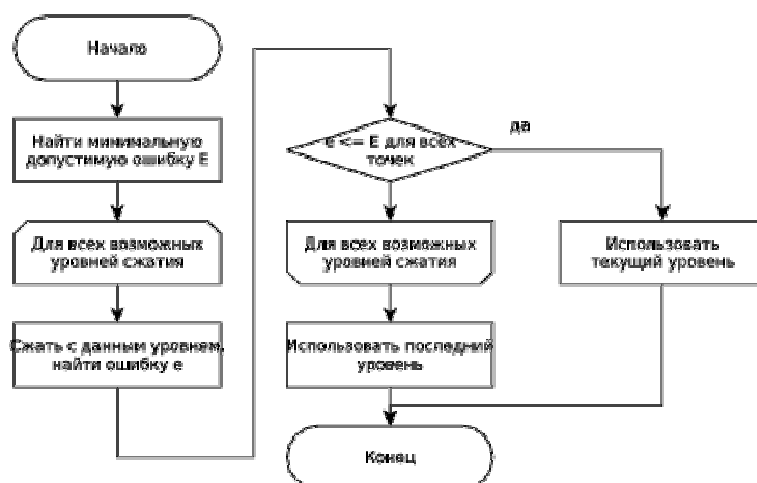


Рис. 7. Схема алгоритма выбора оптимального сжатия

Для выбора палитры в целом, как уже было сказано выше, подходят любые алгоритмы кластеризации. Размер палитры определяется исходя из заданных

возможных уровней сжатия с помощью алгоритмов, представленных выше. Однако когда размер уже известен, необходимо выбрать точки палитры так, чтобы предложенная нами метрика ошибки была минимальной.

Рассматривались несколько алгоритмов:

Выбор средних:

последовательность делилась по средней точке на две части рекурсивно, и от каждой части искалась часть палитры. Так последовательность делится пока на очередном шаге необходимая длина палитры не будет длины 1 – тогда в качестве палитры берётся средняя точка.

K-means:

начальная палитра выбирается любым другим методом. Затем полученная палитра уточняется с помощью алгоритма ближайших соседей.

2-палитра:

рекурсия вглубь проходит так же, как в методе выбора средних. Однако здесь есть дополнительная основа рекурсии на длине палитры 2, когда в качестве палитры берутся крайние точки последовательности по значению.

Статистика уровней сжатия для всех трёх методов выбора палитры показана в таблице 2. Из визуального сравнения видно, что метод K-means позволяет резко улучшить качество изображения при использовании только двух уровней для наименьших блоков размером 3x3.

Таблица 2

Статистика для разных алгоритмов выбора палитры

Уровень	Средние	K-means	2-палитра
1	11%	11%	4%
2	12%	16%	24%
3	45%	45%	23%
4	32%	28%	49%

Возможным этапом постобработки изображения является “сглаживание” ошибок определения палитры (dithering). Это семейство алгоритмов, которые используют особенность человеческого зрения меньше замечать случайные отклонения чем постоянные искажения для улучшения изображения, сжатого с определённой палитрой.

Мы используем разновидность таких алгоритмов, называемую “рапределением ошибок” (error diffusion). Известной разновидностью таких алгоритмов является, например, алгоритм Флойда-Штейнберга [6]. Мы используем модификацию этого алгоритма, описанную в [7]. Общая идея алгоритма сводится к прохождению изображения слева направо, сверху вниз и вычисления разницы между интенсивностью исходного изображения и сжатого в данной точке. Эта ошибка передаётся с различными коэффициентами на окружающие точки. В применении к предложенному формату необходима специальная модификация алгоритма, работающая с различными палитрами в разных областях изображения.

Обработка изображения этим алгоритмом занимает относительно много времени. Результат не отличается по степени сжатия (используется API для модификации изображения без расжатия), однако обладает чуть лучшими визуальными свойствами.

Как описано выше, формат предполагает произвольное количество произвольных уровней сжатия. Нахождение оптимальных уровней аналитическим путём сводится к нахождению оптимального значения многомерной функции. Однако, по времени работы такое нахождение неприемлемо – к тому же функция нелинейна и результат будет оптимален лишь для одного конкретного изображения. Однако, на основе некоторых эмпирических наблюдений и статистики можно построить набор оптимальных уровней сжатия, дающих хороший баланс качества и сжатия.

Для начала определим, какими свойствами должен обладать хороший набор настроек сжатия:

- Все уровни должны использоваться (процент использования в изображениях в среднем относительно качества сжатия и положения среди уровней должен быть приемлем);
- Наиболее сильный уровень сжатия должен давать приемлемые по качеству результаты, если всё изображение будет сжато исключительно им;
- Наиболее слабый уровень должен обладать размером, способствующим равномерному выбору уровней сжатия;
- Количество уровней сжатия должно быть степенью двойки.

Поясним эти требования. Первое требование разумно само по себе, поскольку если уровень не используется в изображениях, он зря занимает место среди остальных. Второе условие исходит из того, что лучший получаемый уровень качества для данного набора уровней – это изображение, целиком сжатое с помощью последнего уровня.

Третье условие получается из того, что индекс уровня кодируется в блоках в двоичной системе счисления, и чтобы полностью заполнить область значений индекса, разумно иметь кратное количество уровней.

Блок 3x3 является, из-за особенностей нашей метрики, удобным для использования. Будем исходить из такого размера как минимального размера блока. Как мы выяснили из визуального эксперимента, блоки 3x3 с двумя уровнями являются оптимальными с точки зрения размер/качество при оптимальном выборе палитры для фотографических и сканированных изображений. Возьмём их за минимум.

При повышении максимального размера блока в два раза наблюдается резкий скачок во время перехода с размера блока 12x12 на 24x24, в котором (для фотографических изображений) качество сжатия сильно ухудшается из-за слишком большой области обхвата, в которой блоки должны быть сжаты одинаковым уровнем сжатия. Таким образом, оптимальным максимальным размером блока было решено выбрать 12x12.

После этого мы изучили различные конфигурации из четырёх уровней сжатия с крайними – сжатием на два уровня по блокам 12x12 и 3x3. Выяснилось, что блок 6x6 с двумя уровнями используется слишком мало. Таким образом, в качестве средних были выбраны блоки 12x12 и 6x6 с четырьмя уровнями. Этот набор обеспечивает стабильное распределение использований уровней с отклонением в сторону слабого сжатия, что соответствует распределению деталей на изображении.

Мы исследовали возможные алгоритмы, применяемые в сжатии и отобрали оптимальные для использования в стандартных настройках сжатия. Мы также нашли метрику, хорошо описывающую ошибку сжатия изображения, и способ нахождения оптимальной степени сжатия при помощи этой метрики. Наконец, мы нашли оптимальную конфигурацию уровней сжатия, обеспечивающую хороший компромисс между сжатием и качеством и равномерно используемую в среднем по различным изображениям.

Список литературы

1. Prof. Bernd Girod S. EE368B – Image and Video Compression. 2001. Available at: <http://web.stanford.edu/class/ee368b>, accessed at 21.11.2014.
2. Duce D. Portable Network Graphics (PNG) Specification (Second Edition): Tech. report. W3C, 2003. Available at: <http://www.w3.org/TR/PNG/> , accessed at 21.11.2014.
3. MSDN. About YUV Video. Microsoft Corporation. Available at:

<http://msdn.microsoft.com/en-us/library/windows/desktop/bb530104%28v=vs.85%29.aspx>, accessed at 21..2014.

4. van den Branden Lambrecht C. Vision Models and Applications to Image and Video Processing. Springer, 2001. ISBN: 9780792374220.
5. Wikipedia. Lempel–Ziv–Markov chain algorithm — Wikipedia, The Free Encyclopedia. 2014. Available at: http://en.wikipedia.org/w/index.php?title=Lempel%E2%80%93Ziv%E2%80%93Markov_chain_algorithm&oldid=630620128, accessed at 14.10.2014.
6. Wikipedia. Floyd–Steinberg dithering — Wikipedia, The Free Encyclopedia. 2014. Available at: http://en.wikipedia.org/w/index.php?title=Floyd%E2%80%93Steinberg_dithering&oldid=608216773, accessed at 23.06.2014.
7. Bloomberg D. S. Color quantization using octrees. Leptonica, 2008. Available at: <http://www.leptonica.com/papers/colorquant.pdf>, accessed at 21.11.2014.