УДК 519.72

Хасанов А. Е., инженер; Юденков В. С., доцент

ПОВЫШЕНИЕ КАЧЕСТВА ЦИФРОВЫХ ИЗОБРАЖЕНИЙ НА ОСНОВЕ РЕКУРСИВНЫХ АЛГОРИТМОВ ИНТЕРПОЛЯЦИИ

Increase of quality of the digital images on a basic recursiving algorithms of interpolation. Rating of quality of the images. Jpeg-algorithm of archiving.

Введение. В настоящее время книгоиздание переживает новую волну подъема: происходит переход от недорогих одноцветных, или дуотоновых, изданий к полноцветным. Это связано с огромной конкуренцией на книжном рынке и повышением запросов потребителей к книжным иллюстрациям. Библиотеки на компакт-дисках не покрывают все темы, поэтому многие издатели ищут иллюстрации в Интернет. В подавляющем большинстве иллюстрации архивированы с потерями и хранятся в формате JPEG. Таким образом, без дальнейшей обработки в программах DTP (Desktop Publisher) иллюстрации чаще всего не пригодны для качественной полноцветной печати. Более того, эти фото в низком разрешении предлагаются для свободного пользования, т. е. бесплатно, что само заставляет искать методы улучшения их качества и включать в публикации.

Оценка качества изображений. Одной из проблем машинной графики является отсутствие адекватного критерия оценки потерь качества изображения при оцифровке, при переводе в ограниченную палитру цветов, при переводе в другую систему цветопредставления для печати и, что для нас особенно важно, при архивации с потерями. Можно привести пример простого критерия качества — среднеквадратичное отклонение значений [1]:

$$d(x, y) = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2}{n^2}}, \qquad (1)$$

где x_{ij} — значение яркости исходного изображения; y_{ij} — значение яркости преобразованного изображения; n — число элементов в строке (столбце) изображения.

Согласно (1), изображение будет сильно испорчено при понижении яркости всего на 5% (глаз этого не заметит — у разных мониторов настройка яркости варьируется гораздо сильнее). В то же время изображения со "снегом" — резким изменением цвета отдельных точек, слабыми полосами, или "муаром", будут признаны "почти не изменившимися". Свои недостатки есть и у других критериев. Рассмотрим, например, критерий максимального отклонения [1]:

$$d(x, y) = \max(x_{ij} - y_{ij}), \qquad (2)$$

где x_{ij} — значение яркости исходного изображения; y_{ij} — значение яркости преобразованного изображения.

Эта мера крайне чувствительна к биению отдельных пикселей, так как во всем изображении может существенно измениться только значение одного пикселя (что практически не заметно для глаза), а согласно (2), изображение будет сильно испорчено.

Мера, которую сейчас используют на практике, называется мерой отношения сигнала к шуму (peak-to-peak signal-to-noise ratio — PSNR) [1].

$$d(x, y) = 10\log 10 \frac{255^2 \cdot n^2}{\sum_{i=1}^{n}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2}, \qquad (3)$$

Данная мера, по сути, аналогична среднеквадратичному отклонению, однако пользоваться ею несколько удобнее за счет логарифмического масштаба шкалы. Ей присущи те же недостатки, что и среднеквадратичному критерию (1).

Лучше всего потери качества изображений оценивают наши глаза. Отличной считается архивация, при которой невозможно на глаз различить первоначальное и разархивированное изображения. Хорошей — когда можно сказать, какое из изображений подвергалось архивации, только сравнивая две находящиеся рядом картинки. При дальнейшем увеличении степени сжатия, как правило, становятся заметны побочные эффекты, характерные для данного алгоритма. На практике, даже при отличном сохранении качества, в изображение могут быть внесены регулярные специфические изменения. Поэтому алгоритмы архивации с потерями не рекомендуется использовать при сжатии изображений, которые в дальнейшем собираются либо печатать с высоким качеством, либо обрабатывать программами распознавания образов. Неприятные эффекты с такими изображениями, как уже говорилось, могут возникнуть даже при простом масштабировании изображения.

JPEG-алгоритм архивации. Первыми для архивации изображений стали применяться привычные алгоритмы [1] — те, что использовались и используются в системах резервного копирования, при создании дистрибутивов и т. п. Эти алгоритмы архивировали информацию без изменений. Однако основной тенденцией в последнее время стало использование новых классов изображений. Старые алгоритмы перестали удовлетворять требованиям, предъявляемым к архивации. Многие изображения практически не сжимались, хотя "визуально" обладали явной избыточностью. Это привело к созданию нового типа алгоритмов — сжимающих с потерей информации. Как правило, коэффициент архивации и, следовательно, степень потерь качества в них можно задавать. При этом достигается компромисс между размером и качеством изображений.

Алгоритм JPEG — один из самых новых и достаточно мощных алгоритмов. Практически он является стандартом де-факто для полноцветных изображений. Оперирует алгоритм областями 8x8, на которых яркость и цвет меняются сравнительно плавно. Вследствие этого при разложении матрицы такой области в двойной ряд по косинусам значимыми оказываются только первые коэффициенты и, таким образом, сжатие в JPEG осуществляется за счет плавности изменения цветов в изображении.

Алгоритм разработан группой экспертов в области фотографии специально для сжатия 24-битных изображений. JPEG — Joint Photographic Expert Group — подразделение в рамках ISO — Международной организации по стандартизации. В целом алгоритм основан на дискретном косинусоидальном преобразовании (ДКП), применяемом к матрице изображения для получения некоторой новой матрицы коэффициентов. Для получения исходного изображения применяется обратное преобразование.

ДКП разлагает изображение по амплитудам некоторых частот. Таким образом, при преобразовании мы получаем матрицу, в которой многие коэффициенты либо близки, либо равны нулю. Кроме того, благодаря несовершенству человеческого зрения, можно аппроксимировать коэффициенты более грубо без заметной потери качества изображения.

Для этого используется квантование коэффициентов. В самом простом случае — это арифметический побитовый сдвиг вправо. При этом преобразовании теряется часть информации, но могут достигаться большие коэффициенты сжатия.

Рассмотрим алгоритм сжатия 24-битного изображения.

Шаг 1.

Переводим изображение из цветового пространства RGB с компонентами, отвечающими за красную (Red), зеленую (Green) и синюю (Blue) составляющие цвета точки, в цветовое пространство YCrCb (YUV) в системе Photoshop — LAB.

Здесь Y — яркостная составляющая, а Cr, Cb — компоненты, отвечающие за цвет (хроматический красный и хроматический синий). За счет того что человеческий глаз менее чувствителен к цвету, чем к яркости, появляется возможность архивировать массивы

для Cr и Cb компонент с большими потерями и, соответственно, большими коэффициентами сжатия. Подобное преобразование уже давно используется в телевидении. На сигналы, отвечающие за цвет, выделяется более узкая полоса частот.

Упрощенно перевод из цветового пространства RGB в цветовое пространство YCrCb можно представить с помощью матрицы перехода:

$$\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} = \begin{vmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.41887 & -0.0813 \\ 0.1687 & -0.3313 & 0.5 \end{vmatrix} * \begin{vmatrix} R \\ G \\ B \end{vmatrix} + \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix}. \quad (4)$$

Обратное преобразование осуществляется умножением вектора YUV на обратную матрицу:

$$\begin{vmatrix} R \\ G \\ B \end{vmatrix} = \begin{vmatrix} 1 & 0 & 1.402 \\ 1 & -0.34414 & -0.71414 \\ 1 & 1.772 & 0 \end{vmatrix} * \left(\begin{vmatrix} Y \\ Cb \\ Cr \end{vmatrix} - \begin{vmatrix} 0 \\ 128 \\ 128 \end{vmatrix} \right). \quad (5)$$

Шаг 2.

Разбиваем исходное изображение на матрицы 8x8 пикселей. Формируем для каждого пикселя полученной матрицы три рабочие матрицы ДКП — по 8 бит отдельно для каждой компоненты. При больших коэффициентах сжатия этот шаг может выполняться чуть сложнее. Изображение делится по компоненте Y, как и в первом случае, а для компонент Cr и Cb матрицы набираются через строчку и через столбец. При этом мы теряем 3/4 полезной информации о цветовых составляющих изображения и получаем сразу сжатие в два раза. Мы можем поступать так благодаря работе в пространстве YCrCb. На обратном RGB изображении, как показала практика, это сказывается незначительно.

Шаг 3.

Применяем ДКП к каждой рабочей матрице. При этом мы получаем матрицу, в которой коэффициенты в левом верхнем углу соответствуют низкочастотной составляющей изображения, а в правом нижнем — высокочастотной.

В упрощенном виде это преобразование можно представить так:

$$Y[u,v] = \frac{1}{4} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} C(i,u) \times C(j,v) \times y[i,j], \quad (6)$$

где

$$C(i,u) = A(u) \times \cos\left(\frac{(2 \times i + 1) \times u \times \pi}{2n} \right). \quad (7, 8)$$

$$A(u) = \begin{cases} \frac{1}{\sqrt{2}} \text{ для } u = 0; \\ 1 \text{ для } u \neq 0. \end{cases}$$

Шаг 4.

Производим квантование. Путем деления рабочей матрицы на матрицу квантования поэлементно. Для каждой компоненты (*Y*, *U* и *V*) в общем случае задается своя матрица квантования $q[u, v]$ (МК).

$$Y_q[u,v] = IntegerRound\left(\frac{Y[u,v]}{q[u,v]} \right). \quad (9)$$

На этом шаге осуществляется управление степенью сжатия и происходят самые большие потери. Понятно, что, задавая МК с большими коэффициентами, мы получим больше нулей и, следовательно, большую степень сжатия.

В стандарт JPEG включены рекомендованные МК, построенные опытным путем. Матрицы для большего или меньшего коэффициентов сжатия получают путем умножения исходной матрицы на некоторое число γ = const.

С квантованием связаны и специфические эффекты алгоритма. При больших значениях коэффициента γ потери в низких частотах могут быть настолько велики, что изображение распадется на квадраты 8x8. Потери в высоких частотах могут проявиться в так называемом "эффекте Гиббса", когда вокруг контуров с резким переходом цвета образуется своеобразный "нимб".

Шаг 5.

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), (1, 1), (0, 2), ...

Таким образом, в начале вектора мы получаем коэффициенты матрицы, соответствующие низким частотам, а в конце — высоким (рис. 1).

Шаг 6.

Свертываем вектор с помощью алгоритма группового кодирования. При этом получаем пары типа (пропустить, число), где "пропустить" является счетчиком пропускаемых нулей, а "число" — значение, которое необходимо поставить в следующую ячейку. Так, вектор 42 3 0 0 0 — 2 0 0 0 0 1 ... будет свернут в пары (0, 42) (0, 3) (3, –2) (4, 1), ...

Шаг 7.

Свертываем получившиеся пары кодированием по Хоффману с фиксированной таблицей.

Процесс восстановления изображения в этом алгоритме полностью симметричен. Метод позволяет сжимать некоторые изображения в 10—15 раз без серьезных потерь.

Положительными характеристиками алгоритма являются:

- задание степени сжатия;
- выходное цветное изображение может иметь 24 бита на точку.

Недостатки алгоритма:

- при повышении степени сжатия изображение распадается на отдельные квадраты (8x8). Это связано с тем, что происходят большие потери в низких частотах при квантовании, и восстановить исходные данные становится невозможно;
- проявляется эффект Гиббса — ореолы по границам резких переходов цветов;
- не очень приятным свойством JPEG является также то, что нередко горизонтальные и вертикальные полосы на дисплее абсолютно не видны и могут проявиться только при печати в виде муарового узора. Он возникает при наложении наклонного растра печати на горизонтальные и вертикальные полосы изображения. Из-за этих свойств JPEG не рекомендуется использовать в полиграфии. Однако при архивации изображений, предназначенных для просмотра человеком, он на данный момент незаменим.

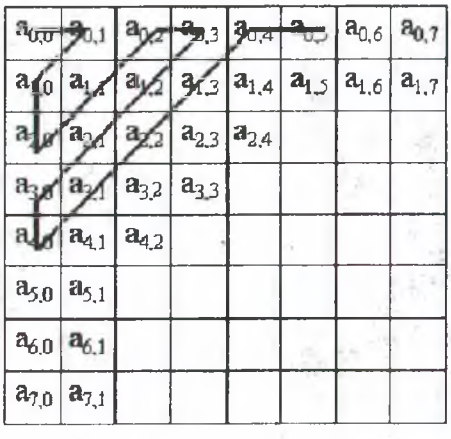

Рис. 1. Схема перевода матрицы 8x8 в 64-элементный вектор методом зигзаг-сканирования

Ликвидация отрицательных явлений JPEG-формата. Интернет-изображения в полиграфическом качестве имеют размер 3—5 см (640x480). Поэтому их необходимо увеличить в 2 раза, чтобы они занимали всю ширину полосы набора в книжных публикациях формата 60x90 $^{1}/_{16}$, 84x108 $^{1}/_{32}$ (A5).

При масштабировании эффект распадания на отдельные квадраты 8x8 особенно превалирует, и это самый существенный недостаток, который есть у JPEG-формата.

Если JPEG-изображение переведено в CMYK-палитру и конвертировано в TIFF-формат, подготовлено к цветоделению, то при детальном рассмотрении видно наличие квадратной регулярной структуры, а если рассматривать поканально, то структура еще более очевидна. Несомненно, что на оттиске это также отпечатается.

Можно попробовать фильтры размытия, это даст какой-то эффект, но полного уничтожения добиться невозможно, кроме того, значительно ухудшится резкость изображения.

Исходя из этого понятно, что нужно работать не со всем изображением, а с границами перехода яркостей областей 8x8, и интерполировать значение яркости на границе раздела.

Алгоритмы интерполяции. Наиболее хорошие результаты дает алгоритм интерполяции Spline (сплайн) [2, 3].

Приближение функции объединением нескольких кривых низкого порядка называется сплайном. Особенно часто используется кубический сплайн. Задача сплайн-интерполирования заключается в построении кусочно-полиномиальной функции, приближающей заданную функцию. Чаще всего приближаемая функция неизвестна, известны лишь измерения этой функции в некоторых точках (узлах).

В нашем случае известны значения яркости в точках соседних квадратов 8x8, необходимо заполнить пространство между ними, чтобы обеспечить плавный переход яркости в некоторой области, что исключит распадание изображения на квадраты.

На рис. 2 приведено исходное изображение фрагмента, на котором явно просматриваются группы пикселей матрицы ДКП для каждой цветовой составляющей. Учитывая симметричность JPEG-преобразования, можно однозначно определить местоположение границ областей 8x8 пикселей и построить битовую маску, которая опишется матрицей M, аналогичной для обеих цветовых составляющих Cr и Cb (u и v). В матрицу войдут элементы 1, 8, 9, 16, 17, … ,т. е. $n + 7$ и $n + 8$, где $n = 1,\ldots, k$, k — число элементов каждой строки (столбца). Для упрощения вычислений первую и последнюю строку, первый и последний столбец изображения можно исключить, их потеря понизит количество информации на 0.2—0.5%, что незначительно. Результат работы приведенного выше преобразования представлен на рис. 3.

Окончательную обработку выделенного фрагмента произведем на основе кубического сплайна, задавая в приведенном ниже алгоритме $K = 3$.

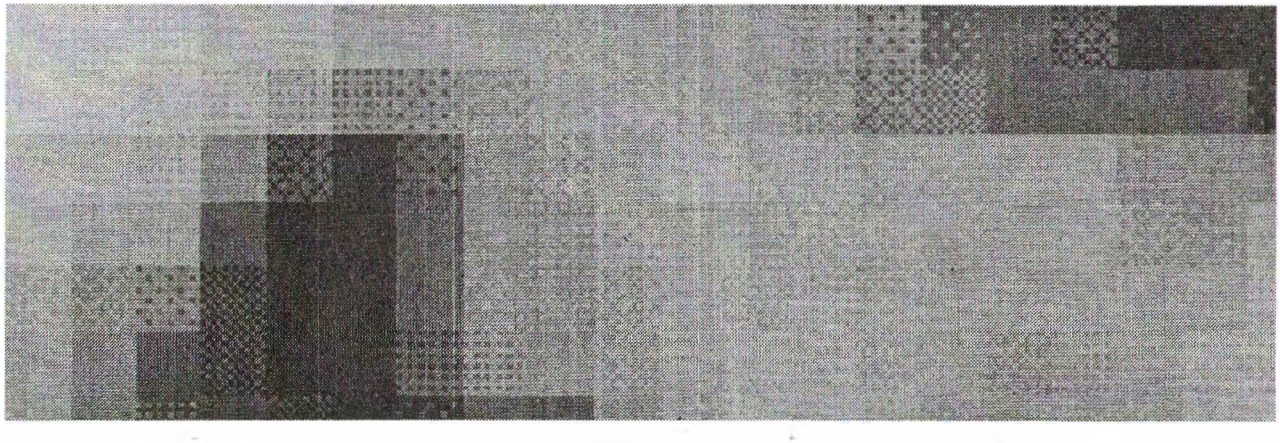

Рис. 2. Группы пикселей 8x8 в матрицах Cr и Cb (u и v)

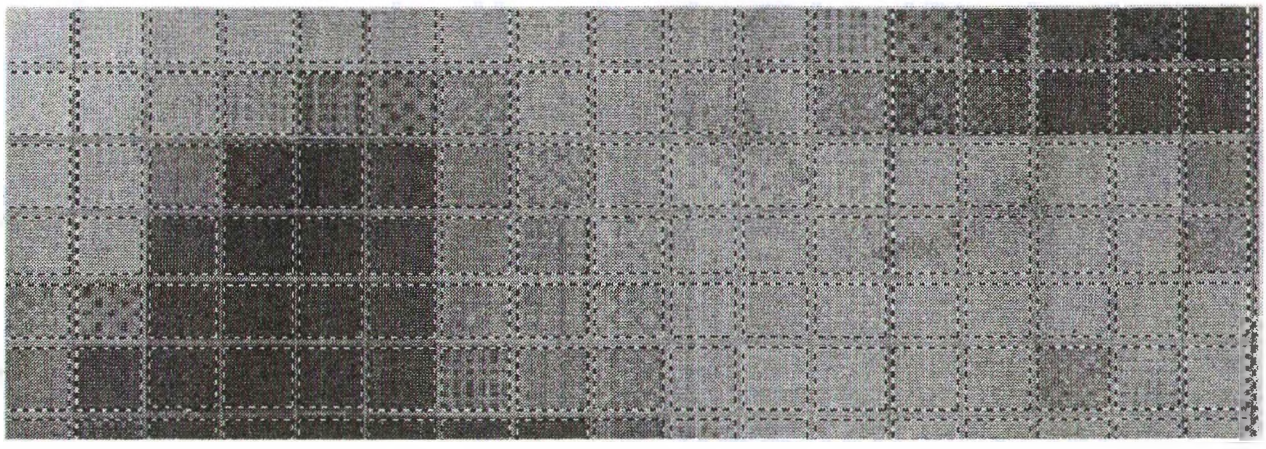

Рис. 3. Битовая маска, скопированная в α-канал

Пусть задана сетка узлов (узлы сплайна): $t_1 \le t_2 \le \ldots \le t_k < t_{k+1} < \ldots < t_n < t_{n+1} \le t_{n+2} \le \ldots \le t_{n+k}$, определяющая для $i = 1, 2, \ldots, n$ нормированные B-сплайны $N_{ik}(x)$ порядка k, соответствующие узлам $t_i, t_{i+1}, \ldots, t_{i+k}$.

Строится сплайн-функция:

$$f(x) = \sum_{i=1}^{n} a_i N_{ik}(x), \qquad t_1 \le x \le t_{n+1}, \tag{10}$$

коэффициенты a_i, $i = 1, 2, \ldots, n$, которой определяются условиями интерполяции

$$\sum_{i=1}^{n} a_i N_{ik}(z_j) = b_j, \quad j = 1, 2, \ldots, n,$$

где $Z_1 < Z_2 < \ldots < Z_n$ — узлы интерполяции, такие, что $[Z_1, Z_n] \subseteq [t_1, t_{n+1}]$; b_j — заданные значения.

Параметры, необходимые для задания сплайн-интерполяции, приведены в таблице.

Таблица

Параметры сплайн-интерполяции

N	заданное число узлов интерполяции (тип: целый);
K	заданный порядок B-сплайна (тип: целый);
T	вещественный вектор длины $N + K$ значений узлов сплайна: $T(1) \le T(2) \le \ldots \le T(K) < T(K+1) < T(K+2) < T(N) < T(N+1) \le T(N+2) \le \ldots \le T(N+K)$;
Z	вещественный вектор длины N значений узлов интерполяции: $T(1) \le Z(1) < Z(2) < \ldots < Z(N) \le T(N+K)$;
A	вещественный вектор длины N: $a_i = A(I)$, $I = 1, 2, \ldots, N$;
B	вещественный вектор длины N: $b_I = B(I)$, $I = 1, 2, \ldots, N$;
R	вещественный двумерный рабочий массив размера $N * (2*K - 1)$;
R1	вещественный рабочий вектор длины K;
R2	вещественный двумерный рабочий массив размера $N*K$;
IERR	целая переменная, служащая для сообщения об ошибках, обнаруженных в ходе работы подпрограммы; при этом:
IERR=65	когда какой-то узел $Z(I) \notin [T(1), T(N+1)]$;
IERR=66	когда матрица для определения коэффициентов вырождена.

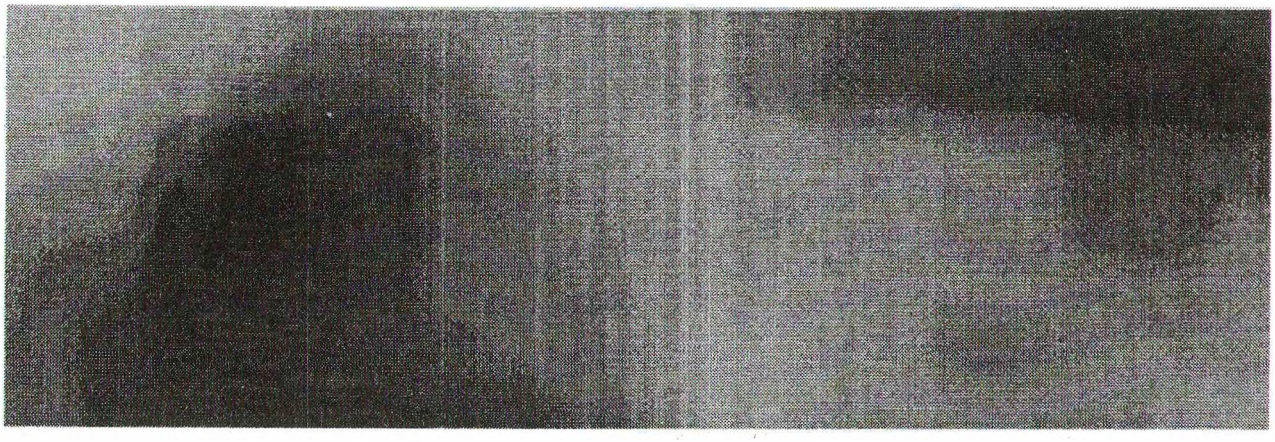

Рис. 4. Результат сплайн-интерполяции

В результате обработки получим сглаженное изображение, приведенное на рис. 4.

Выводы. На основе анализа алгоритма JPEG-преобразования выявлена закономерность появления перепадов яркости на границах областей 8×8 пикселей. На основе выявленной закономерности построена битовая маска для изображения. Применив к полученной битовой маске кубическую сплайн-интерполяцию сохраним четкость образов и получим возможность масштабировать JPEG-изображения, исключая недостатки алгоритма сжатия.

Для управления степенью фильтрации можно ввести управление шириной битовой маски и выбором опорных точек интерполяции.

ЛИТЕРАТУРА

1. Ватолин Д., Ратушняк А., Смирнов М., Юкин В. Методы сжатия данных. — М.: Наука, 2003.

2. Ахмеров Р.Р., Коробицына Ж.Л., Слепцов А.Г. Основы численного анализа в задачах. — Новосибирск: Изд-во НГУ, 1994.

3. Барахнин В.Б., Шапеев В.П. Введение в численный анализ. — Новосибирск: Изд-во НГУ, 1997.