

ИССЛЕДОВАНИЕ МЕТОДОВ ОПТИМИЗАЦИИ АЛГОРИТМА JPEG

А.И. Стрелков¹, В.И. Барсов², А.В. Воронин²

(¹Харьковский университет Воздушных Сил,

²Украинская инженерно-педагогическая академия, Харьков)

Рассмотрены математические модели алгоритмов JPEG сжатия на предмет оптимизации классической схемы сжатия, предложены новые методы оптимизации, позволяющие получить более высокие результаты по сравнению с классикой. Предложены критерии оценки методов оптимизации.

алгоритмы JPEG, классическая схема сжатия, методы оптимизации

Постановка проблемы. В статье «Исследование влияния случайного шума на степень сжатия изображения по алгоритму JPEG»[1] были предложены критерии оценки алгоритмов сжатия изображений на предмет устойчивости к помехам случайного характера, и рассматривалась возможность создания оптимизированного алгоритма, устойчивого к воздействию помех. Исследовалось сжатие по алгоритму JPEG, который в своей основе содержит дискретное косинусное преобразование. Исследования показали, что данный алгоритм, при сильно зашумленном изображении и сжатии без потерь не только не сжимает, но и увеличивает файл изображения. А при сжатии с потерями изображение очень сильно страдает в качественной характеристике.

Для дальнейшей разработки помехозащищенных алгоритмов необходимо рассмотреть, чем же помехи так влияют на алгоритм. С точки зрения рациональности было бы не уместным разрабатывать новый алгоритм, основываясь на алгоритме JPEG. Рассмотрим алгоритм на предмет оптимизации его работы.

Анализ литературы. Алгоритм сжатия цифровых изображений JPEG [2] на сегодняшний день является фактическим стандартом, поддерживаемым многочисленными программными и аппаратными системами цифровой обработки, хранения и передачи изображений. Ограничимся случаем обработки черно-белых полутоновых изображений и рассмотрим кратко основную схему кодирования (sequential encoding) спецификации JPEG, которая наиболее широко применяется на практике. Спецификация JPEG состоит из нескольких частей, включая сжатие с

потерями и без. Далее в этой статье речь пойдет о части JPEG, реализующей сжатие с потерями, ведь как показано в статье «Исследование влияния случайного шума на степень сжатия изображения по алгоритму jpeg»[1], шум влияет на отношение сигнал/шум только если имеют место потери, т.е. используются алгоритмы сжатия с потерями.

При работе компрессора JPEG последовательно выполняются следующие три операции (рис. 1).

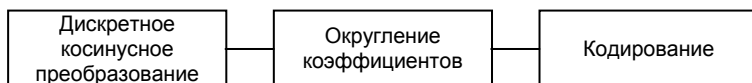


Рис. 1. Последовательность работы алгоритма JPEG

Эти три ступени сжатия образуют мощный компрессор, способный сжимать графические образы примерно в десять раз, теряя очень немного в качестве изображения.

Ключевым компонентом работы алгоритма является дискретное косинусное преобразование [3] (далее ДКП). ДКП представляет собой разновидность преобразования Фурье и, так же как и оно, имеет обратное преобразование (ОДКП). Графическое изображение можно рассматривать как совокупность пространственных волн, причем оси X и Y совпадают с шириной и высотой картинка, а по оси Z откладывается значение цвета соответствующего пиксела изображения. ДКП позволяет переходить от пространственного представления картинка к ее спектральному представлению и обратно. Воздействуя на спектральное представление картинка, состоящее из «гармоник», то есть отбрасывая наименее значимые из них, можно балансировать между качеством воспроизведения и степенью сжатия.

ДКП преобразует матрицу пикселей размером $N \times N$ в матрицу частотных коэффициентов соответствующего размера:

$$DCT(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \text{pixel}(x, y) \cos \left[\frac{(2x+1)i\pi}{2N} \right] \cos \left[\frac{(2y+1)j\pi}{2N} \right]; \quad (1)$$

$$C(x) = \begin{cases} \frac{1}{\sqrt{2}}, & x = 0; \\ 1, & x > 0. \end{cases} \quad (2)$$

Несмотря на видимую сложность, закодировать эти формулы достаточно просто.

Нетрудно заметить, что время, необходимое для вычисления каждого элемента матрицы ДКП, сильно зависит от ее размера. Так как ис-

пользуются два вложенных цикла, время вычислений составляет $O(N \times N)$. Одним из наиболее болезненных последствий является то, что практически невозможно выполнить ДКП всего изображения сразу. В качестве решения этой проблемы группа разработчиков JPEG предложила разбивать изображение на блоки размером 8×8 точек[3]. Увеличивая размеры блока ДКП, можно добиться улучшения результатов сжатия, но не до бесконечности, так как мало вероятно, что сильно удаленные точки «похожи» друг на друга.

$$C(i, j) = \begin{cases} \frac{1}{\sqrt{N}}, & i = 0; \\ \frac{\sqrt{2}}{N} \cos \left[\frac{(2j+1)i\pi}{2N} \right], & i > 0. \end{cases} \quad (3)$$

По определению ДКП для его реализации потребуется два вложенных цикла. Тело циклов будет выполняться $N \times N$ раз для каждого элемента матрицы ДКП. Значительно более эффективный вариант вычисления коэффициентов ДКП может быть реализован через перемножение матриц [4].

Формула ДКП может быть записана несколько по-другому:

$$\text{ДКП} = \text{КП} \times \text{Точки} \times \text{КПт}. \quad (4)$$

Здесь КП – матрица косинусного преобразования размером $N \times N$, элементы которой определяются по формуле (3); точки – матрица размером $N \times N$, состоящая из точек картинки; КПт – транспонированная матрица КП; оператор обозначает умножение матриц. Если рассматривать алгоритм JPEG и приведенную программу, то размер всех матриц составляет 8×8 элементов. При перемножении матриц «цена» вычисления одного элемента результирующей матрицы составляет N умножений и N сложений, при вычислении матрицы ДКП – $2 \times N$ соответственно. По сравнению с $O(N \times N)$ это заметное повышение производительности.

Таким образом, ДКП представляет собой преобразование информации без потерь и не осуществляет никакого сжатия. Напротив, ДКП подготавливает информацию для этапа сжатия с потерями или округления.

Округление представляет собой процесс уменьшения количества битов, необходимых для хранения коэффициентов матрицы ДКП за счет потери точности.

Стандарт JPEG реализует эту процедуру через матрицу округления. Для каждого элемента матрицы ДКП существует соответствующий элемент матрицы округления. Результирующая матрица получается делени-

ем каждого элемента матрицы ДКП на соответствующий элемент матрицы округления и последующим округлением результата до ближайшего целого числа [3]. Как правило, значения элементов матрицы округления растут по направлению слева-направо и сверху-вниз.

Очевидно, что от выбора матрицы округления зависит баланс между степенью сжатия изображения и его качеством после восстановления. Стандарт JPEG позволяет использовать любую матрицу округления, однако ISO разработала набор стандартных матриц для округления. Эти матрицы были получены в результате длительного тестирования членами JPEG и образуют прочную основу для дальнейших экспериментов.

Следовательно, операция округления является единственной фазой работы JPEG, где происходит потеря информации.

Целью работы является выявление причин влияния сжатия JPEG на качество зашумленного изображения, при условии сжатия с потерями, разработка методов уменьшения степени влияния.

Основная часть. В самом общем виде оптимизацию произвольного алгоритма кодирования данных с потерями можно свести к нахождению минимума функции

$$J(u) = D(u) + \lambda R(u), \quad (5)$$

где D – мера вносимой ошибки; R – длина битового кода; параметр $\lambda \geq 0$ определяет необходимое соотношение между качеством обработки и длиной кода, а вектор u представляет собой набор параметров, которые позволяет варьировать конкретный алгоритм. В качестве меры вносимых искажений обычно используется среднеквадратичная ошибка (PSNR). Предлагаемый нами метод оптимизации кодера JPEG может быть представлен как процесс минимизации функции (5), где в качестве варьируемых параметров и используются:

- набор из 100 рекомендуемых JPEG матриц квантования, $\{Q_Q\}_{Q=1}^{100}$ (чем больше индекс матрицы, тем меньше ошибка квантования);
- все возможные таблицы кодов Хаффмена $\{h\}$;
- все возможные способы дополнительного усечения коэффициентов спектров, $L = \bigcup L_i$.

Таким образом, $u = \{Q, h, L\}$.

Предлагаемый алгоритм оптимизации кодера JPEG, т.е. алгоритм минимизации функции (5) при заданном λ , можно разбить на три следующих шага.

1. Проинициализировать таблицу кодов Хаффмана по таблице, ре-

комендуемой группой разработчиков JPEG, $h \leftarrow h_{\text{JPEG}}$.

2. Минимизировать по Q функцию

$$J(Q, h) = \min_L [J(Q, L, h) = D(Q, L) + \lambda R(Q, L, h)], \quad (6)$$

т.е. определить $Q^* = \arg \min_{Q \in \{Q_1, \dots, Q_{100}\}} J(Q, h)$. При этом находим также L^* .

3. Выполнить минимизацию функции (2) (или (1)) по h , т.е. определить

$$h^* = \arg \min [J(Q^*, L^*, h) = D(Q^*, L^*) + \lambda R(Q^*, L^*, h)].$$

Считаем оптимальные параметры (Q^*, L^*, h^*) найденными. Процедура поиска оптимальных кодов Хаффмана h^* не является итерационной (шаг 3), что практически не влияет на характеристики оптимизации, однако позволяет использовать быстрые алгоритмы поиска на шаге 2.

Процедуру усечения коэффициентов спектров представим как результат двух воздействий: обнуления T и уменьшения C абсолютных величин (до ненулевого значения), $L = T \times C$. Если способ усечения L_i не изменяет коэффициентов i -го спектра, будем обозначать это как $L_i = 1$ (при этом, очевидно $T_i = C_i = 1$).

Поиск $L^* = T^* \times C^*$ (7) производится в два этапа.

1. Минимизировать функцию (5) по T , считая $C = 1$, т.е. определить

$$T^* = \arg \min [J(Q, T \times 1, h) = D(Q, T \times 1) + \lambda R(Q, T \times 1, h)].$$

2. Минимизировать функцию (5) по C , т.е. определить

$$C^* = \arg \min [J(Q, T^* \times C, h) = D(Q, T^* \times C) + \lambda R(Q, T^* \times C, h)].$$

Поиск оптимального усечения проводится для каждого фрагмента по отдельности, т.е.

$$L^* = \bigcup L_i^* (T^* = \bigcup T_i^*, C^* = \bigcup C_i^*).$$

Особенности выполнения этапа 1, способы построения быстрых процедур поиска T^* – подробно рассмотрены в работе [5]. Для оставшихся после первого этапа ненулевых коэффициентов $\tilde{y} \neq 0$ далее на втором этапе необходимо проверять возможность их перевода в каждый из диапазонов меньшей разрядности, при этом ясно, что возможность полного обнуления коэффициентов исключена предшествующим этапом оптимизации по T

(поэтому рассматривать значения $\tilde{y} = \pm 1$ не имеет смысла).

Однако перенос проквантованного коэффициента дальше, чем в ближайший диапазон — это уже «почти обнуление» по величине вносимой ошибки, и в силу того, что возможность обнуления была проанализирована ранее, можно ограничиться анализом только одного варианта дополнительного усечения — на общую эффективность оптимизации такое упрощение не оказывает заметного влияния, однако сокращает вычислительные затраты.

Выводы. В данном случае, оптимизация алгоритма проведена в целом не затрагивая округление. Дополнительный выигрыш в PSNR невелик и на тестовых изображениях составляет величину около 0,1 дБ, однако соответствующее усложнение оптимизации не влечет за собой практически никаких вычислительных затрат.

Перспективы дальнейших исследований. Оптимизацию сжатия по алгоритму JPEG актуально проводить на этапе округления коэффициентов, так как оптимизация на других этапах не дает значительного выигрыша в PSNR, данный факт был установлен в результате проведенных исследований.

ЛИТЕРАТУРА

1. Стрелков А.И., Барсов В.И., Воронин А.В. Исследование влияния случайного шума на степень сжатия изображения по алгоритму JPEG // Системы обработки информации. – X.: ХВУ, 2004. – Вып. 12 (40). – С. 218 – 221.
2. Storer J.A. *Data Compression: Methods and Theory*. – USA: Computer Science Press, 1988.
3. Nelson M. *The Data Compression Book*. – USA: M&T Publishing, 1991.
4. Wallace G.K. *The JPEG Still Picture Compression Standard // Communication of the ACM*. – 1991. – Vol. 34, № 4.
5. Мастрюков Д. Алгоритмы сжатия информации: Ч. 5. Алгоритмы сжатия в драйверах устройств // Монитор. – 1994. – № 4.
6. Wallace G.K. *The JPEG algorithm for image compression standard // Communications of the ACM*. – 1991. – Vol. 34, № 4. – P. 30 – 44.
7. Crouse M., Ramchandrn K. *Joint thresholding and quantizer selection for transform image-coding: entropy-constrained analysis and applications to baseline JPEG // IEEE Trans, on Image Processing*. – 1997. – Vol. 6, № 2. – P. 285 – 297.

Поступила 31.08.2005

Рецензент: доктор технических наук, профессор Ю.В. Стасев,
Харьковский университет Воздушных Сил.