

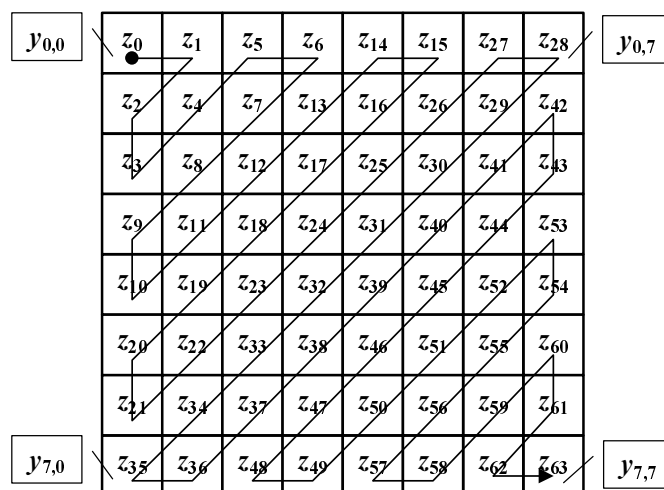
Использование контекстного арифметического кодирования для повышения сжатия данных по схеме JPEG

С.В.Умняшкин,

Московский государственный институт электронной техники

Предложен алгоритм арифметического кодирования коэффициентов дискретного косинусного преобразования, ориентированный на применение для цифровой компрессии изображений. По сравнению со стандартной схемой JPEG эффективность сжатия данных повышается на 10%.

Рассмотрим схему компрессии цифровых изображений, при которой изображение обрабатывается фрагментами 8×8 пикселей при помощи двумерного дискретного косинусного преобразования (ДКП) [1]. Пусть матрица $\mathbf{X} = \{x_{k,l}\}_{k,l=0}^7$ - результат ДКП, а матрица $\mathbf{Y} = \{y_{k,l}\}_{k,l=0}^7$ получена после равномерного квантования коэффициентов матрицы \mathbf{X} : $y_{k,l} = \text{round}(x_{k,l}/q_{k,l})$, где $\mathbf{Q} = \{q_{k,l}\}_{k,l=0}^7$ - целочисленная матрица. Обозначим $\mathbf{Z} = (z_0, \dots, z_{63})$ - вектор, полученный в результате зигзагообразного считывания матрицы \mathbf{Y} в одномерную последовательность, см. рис. Последовательность векторов $\{\mathbf{Z}_j\}$, соответствующих отдельным фрагментам изображения, образует данные, которые далее подвергаются статистическому кодированию. Такой способ обработки соответствует подходу, применяемому в методах MPEG и JPEG [1].



Рассмотрим возможность обработки последовательности векторов $\{\mathbf{Z}_j\}$ с помощью алгоритма арифметического кодирования [2], использующего несколько статистических моделей, которые адаптируются в процессе обработки под статистику соответствующих им потоков данных. Так же, как и в методе JPEG, кодировать будем ненулевые значения компонент вектора \mathbf{Z} и длины серий разделяющих их нулей. Выделим следующие потоки данных, каждому из которых

соответствует своя отдельная статистическая модель (т.е. накопленная по уже обработанным данным гистограмма частот появления различных символов).

- Поток D , соответствующий постоянным составляющим яркости фрагментов $z_0 = y_{0,0}$.
- N_0 потоков $\{S_i\}_{i=1}^{N_0}$, в которых кодируются длины нулевых серий.
- N_1 потоков $\{C_i\}_{i=1}^{N_1}$, в которых кодируются ненулевые компоненты.

Таким образом, всего используется (N_0+N_1+1) потоков данных $\{T\}$. Сначала в потоке D постоянные составляющие яркости кодируются в виде разностей значений, соответствующих соседним фрагментам изображения. Затем по следующему правилу происходит кодирование компонент z_1, \dots, z_{63} .

Шаг 0. (Подготовительный).

- 0.1.** Определить индекс j_{\max} последнего ненулевого элемента $z_{j_{\max}} \neq 0, \forall k > j_{\max} : z_k = 0$. (Если $z_1 = \dots = z_{63} = 0$, то $j_{\max} = 0$.)
- 0.2.** Индекс текущей обрабатываемой компоненты $z_j: j \leftarrow 0$.
- 0.3.** Установить текущий поток (модель) $T \leftarrow S_{N_1}$.
- 0.4.** Длина нулевой серии: $n_z \leftarrow 0$.

Шаг 1. $j \leftarrow j+1$. Если $j > j_{\max}$, то перейти на шаг 3.

Шаг 2. Если $z_j = 0$ то

- 2.1.** Длина нулевой серии: $n_z \leftarrow n_z + 1$. Перейти на шаг 1.
иначе
- 2.2.** В текущем потоке T закодировать n_z ; положить $n_z \leftarrow 0$.
- 2.3.** Вычислить контекстный прогноз $p = p_j(z_1, \dots, z_{j-1})$.
- 2.4.** Установить текущий поток $T \leftarrow C_k$. Номер $k, 1 \leq k \leq N_1$, находится по условию: $p \in [t_{k-1}^C, t_k^C)$, где $\{t_m^C\}_{m=0}^{N_1}$ - некоторые пороги, $0 = t_0^C < t_1^C < \dots < t_{N_1}^C = \infty$.
- 2.5.** В потоке T закодировать компоненту z_j .
- 2.6.** Установить текущий поток (модель) $T \leftarrow S_k$. Номер $k, 1 \leq k \leq N_0$, находится по условию: $p \in [t_{k-1}^S, t_k^S)$, где $\{t_m^S\}_{m=0}^{N_0}$ - пороги, $0 = t_0^S < t_1^S < \dots < t_{N_0}^S = \infty$.
- 2.7.** Перейти на шаг 1.

Шаг 3. Если $j < 64$ то закодировать в потоке T признак конца блока ЕОВ.

Как видно из приведенного описания алгоритма кодирования, ключевую роль в его работе играет правило построения прогнозной величины $p = p_j(z_1, \dots, z_{j-1})$ по контексту уже закодированных компонент, сводится к вычислению взвешенной суммы абсолютных значений

закодированных компонент:

$$p_j = \sum_{k=1}^j \alpha_{j,k} |z_k|, \quad (1)$$

где весовые коэффициенты $\alpha_{j,k} \geq 0$. Эксперименты показывают, что для составления прогноза p_j (1) достаточно брать 2-3 коэффициента вектора, которые являются ближайшими закодированными соседями в двумерном спектре ДКП, см. рисунок. Например, для коэффициента z_3 : $p_3 = 0.32|z_1| + 0.81|z_2|$, а для z_4 прогноз имеет вид:

$$p_4 = |z_1| + |z_2|. \quad (2)$$

Конкретные значения весов $\alpha_{j,k}$ определяются эмпирически, по результатам обработки тестовых изображений.

Отметим также особенность, связанную с составлением контекстного прогноза для компонент z_1 и z_2 . Контекст первых коэффициентов включает в себя также прогнозную величину \hat{p}_4 (2), вычисленную по предыдущему вектору $\hat{\mathbf{Z}}$, кодирование которого предшествовало обработке текущего вектора \mathbf{Z} : $p_1 = 3\hat{p}_4$, $p_2 = 1.2(\hat{p}_4 + |z_1|)$.

Другими важными параметрами алгоритма являются значения порогов $\{t_m^C\}_{m=0}^{N_1+1}$ и $\{t_m^S\}_{m=0}^{N_0+1}$, используемые для получения номера статистической модели в результате квантования прогнозных величин p_j на шагах 2.4 и 2.6. Универсальные, пригодные для обработки различных изображений пороги были определены по результатам экспериментов, так же, как и весовые коэффициенты $\alpha_{j,k}$. При этом для кодирования длин нулевых серий и для кодирования ненулевых элементов следует выбирать по четыре статистических модели, т.е. $N_0 = N_1 = 4$. Результаты применения описанного контекстного прогнозирования для адаптивного арифметического кодирования показывают, что по сравнению со стандартной схемой JPEG величина сжатия данных возрастает примерно на 10%. В качестве эталона для определения характеристик сжатия по стандартной схеме JPEG использовалась программа JPEG OptimizerTM версии 4.0 (см. <http://xat.com>).

Отметим, что дополнительные вычисления прогнозов вида (1), необходимые для реализации описанного алгоритма, не влекут драматического усложнения вычислений, так как расчет прогнозных значений требуется только при появлении в векторе \mathbf{Z} ненулевой компоненты, реально же подавляющее большинство компонент вектора – нули. Кроме того, прогнозы (1) составляются только из 2-3 слагаемых, причем многие весовые коэффициенты получаются целыми, или даже единичными – все это существенно упрощает реализацию необходимых арифметических операций на реальной вычислительной аппаратуре.

В заключение отметим, что рассмотренный подход к использованию адаптивного арифметического кодирования при обработке коэффициентов ДКП в JPEG-подобной схеме компрессии

изображений касается только итоговой фазы статистического кодирования, не вносящей потерь данных. Дополнительно повысить эффективность всей схемы компрессии можно, если помимо изменения этапа статистического кодирования применить схему оптимизации квантования по аналогии с тем, как это выполняется для стандартной схемы JPEG [3,4].

Литература

1. *Wallace G.K.* The JPEG algorithm for image compression standard // Communications of the ACM. - 1991. - Vol. 34. - №4. - P. 30-44.
2. *Witten I., Neal R.M., Cleary J.G.* Arithmetic coding for data compression // Comm. ACM. - 1987. - V.30. - №6. – pp. 520-540.
3. *M.Crouse and K.Ramchandran.* 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.
4. *Умняшкин С.В., Космач М.В.* Оптимизация кодирования цифровых изображений по методу JPEG //Известия вузов. Электроника. - №4-5. - 2000. - С. 139-141.