

Эффективное сжатие изображений с использованием обобщенных интервальных преобразований

И.В. Браиловский

Введение

Одним из наиболее распространенных подходов к задаче сжатия изображений без потерь качества является семейство методов кодирования ошибки предсказания. Эти методы объединяет то, что общая схема работы подобных алгоритмов может быть разбита на две фазы: фазу моделирования и фазу кодирования. На этапе моделирования с помощью модели некоторого класса изображений производится оценка данных текущего изображения, позволяющая предсказывать значение текущего значения яркости точки по предыдущим данным. На этапе кодирования производится представление ошибки предсказания, иначе говоря, разности между реальным значением яркости и предсказанным в некоторое сжатое (обычно двоичное) представление. Такая схема сжатия изображений впервые была представлена в алгоритме Sunset [2]. Существуют различные вариации этого метода, так, например, существуют однопроходные методы, например, в стандарте JBIG [3] или алгоритме FELICS [4,5], когда ошибка предсказания сразу кодируется, или двухпроходные, когда сначала вычисляются все ошибки, а на втором проходе производится кодирование. Например, в стандарте Lossless JPEG [6] используются оба варианта: однопроходный, когда ошибка предсказания кодируется сразу, и двухпроходный, когда ошибка предсказания на втором проходе кодируется с помощью арифметического кодирования. Рассматриваются также различные типы входных изображений: двухцветные, черно-белые с оттенками серого цвета, цветные. Естественно, эти методы характеризуются различными степенями сжатия, различной алгоритмической сложностью и т.д. Одним из наилучших методов по соотношению алгоритмической сложности и эффективности сжатия является известный стандарт компрессии полутоновых изображений без потерь качества JPEG-LS [1,7,8].

Применение обобщенных интервальных в задаче сжатия изображений естественно рассматривать применительно ко второй фазе методов, основанных на предсказании, - к фазе кодирования. Далее применение интервальных преобразований будет рассматриваться на примере JPEG-LS, хотя аналогичное применение может иметь место и в случае других алгоритмов. JPEG-LS превосходит по степени сжатия многие другие алгоритмы своего класса сложности на фотореалистичных полутоновых изображениях. Однако, на синтезированных полутоновых изображениях, таких как, например, картографические изображения, мультипликационные изображения, интернет-графика и т.д., JPEG-LS несколько уступает отдельным алгоритмам. Применение обобщенных интервальных преобразований, как далее будет показано, позволяет повысить степень сжатия искусственных изображений, при этом алгоритмическая сложность такого модифицированного метода практически не изменяется. Наличие модификации JPEG-LS, способной эффективно сжимать искусственные изображения, позволяет расширить применение этого алгоритма на широкий класс изображений, используемый в различных практических областях – например, в ГИС системах (спутниковые изображения земной поверхности, картографические изображения и т.д.) или медицинских системах (рентгеновские фотоснимки, изображения кардиограмм и т.д.).

Основой стандарта JPEG-LS является алгоритм LOCO-I (Low COmplexity LOssless COmpression for Images). Схема работы этого алгоритма, по существу, не отличается от общей схемы методов кодирования ошибки предсказания, однако есть некоторые важные уточнения. Во-первых, следует отметить, что LOCO-I работает за один проход, т. е. для каждой точки сначала вычисляется ошибка предсказания и затем сразу эта ошибка кодируется. Во-вторых, JPEG-LS может работать с разными типами изображений, в том числе и цветными, но при этом кодируя каждую цветовую плоскость отдельно. Поэтому мы можем ограничить дальнейшее рассмотрение на черно-белые изображения с оттенками серого цвета. Далее, более точно процесс предсказания и кодирования значения яркости x текущей точки можно разбить на три шага:

Шаг 1: предсказание значения \hat{x} яркости текущей точки по предыдущим значениям.

Шаг 2: вычисление контекста текущей точки, который, в свою очередь, является также функцией предыдущих значений яркости.

Шаг 3: по полученным на шаге 1 и 2 данным вычисляются параметры вероятностной модели для ошибки предсказания $\hat{\mathcal{E}} = \hat{x} - x$ и производится кодирование этой ошибки.

Для вычисления предсказанного значения \hat{x} яркости текущей точки по предыдущим данным в LOCO-I предлагается использовать соседние к x точки, так, как показано на Рис. 1. Сам фильтр для предсказания значения \hat{x} , согласно стандарту JPEG-LS, может быть любым, однако по умолчанию используется фильтр MED (Median Edge Detector):

$$\hat{x} = \begin{cases} \min(a, c), & \text{если } b \geq \max(a, c) \\ \max(a, c), & \text{если } b \leq \min(a, c) \\ a + c - b, & \text{иначе} \end{cases}$$

В качестве контекста используются упорядоченные номера значений троек следующих величин: $g_1 = d - c$, $g_2 = c - b$ и $g_3 = b - a$. Иначе говоря, мы нумеруем множество всех возможных значений (с точностью до некоторой симметрии) вектора g_1, g_2, g_3 и по текущим значениям вычисляем контекст.

*	*	*	*	*
*	b	c	d	*
*	a	x	*	*

Рис. 1. Взаиморасположение опорных точек изображения для предсказания текущего значения яркости

Относительно ошибки $\hat{\mathcal{E}} = \hat{x} - x$ предсказания принимается общепризнанная гипотеза [9] о том, что она имеет распределение Лапласа, т.е. симметричное относительно некоторого значения экспоненциально убывающее распределение. С учетом того, что вероятностная модель применяется для каждого контекста в отдельности, удается сместить центр симметрии распределения так, чтобы он находился достаточно близко к нулю:

$$P_\rho(\hat{\mathcal{E}}) = C(\rho, p) p^{|\hat{\mathcal{E}} + \rho|},$$

где $\hat{\mathcal{E}} = 0, \pm 1, \pm 2, \dots$ - ошибка предсказания, $0 < p < 1$, $\rho < 1$ - некоторый параметр, характеризующий центр симметрии распределения, а $C(\rho, p)$ - нормирующий множитель, выбираемый так, чтобы $P_\rho(\hat{\mathcal{E}})$ образовывали распределение вероятностей, т.е.:

$$\sum_{\hat{\mathcal{E}}} P_\rho(\hat{\mathcal{E}}) = 1.$$

Для случая искусственных изображений вероятность значения “ноль” ошибки предсказания значительно выше, чем для фотореалистичных изображений; при этом, как видно из формулы для $P_\rho(\hat{\mathcal{E}})$, остальные значения ошибки имеют исчезающе малые вероятности. Кодируется $\hat{\mathcal{E}}$ с помощью посимвольного кодирования, основанного на коде Райса-Голомба, и оптимального для распределения вида $P_\rho(\hat{\mathcal{E}})$ [8]. В случае синтезированных изображений вероятность серий из нулей оказывается достаточно большой. При кодировании каждого нуля с помощью посим-

вольного кода типа Райса-Голомба для серии длиной N будет использовано не менее N бит. Однако, при использовании кодирования длин серий можно затрачивать немногим более $\log_2 N$ бит на всю серию из N нулей. Это наблюдение и служит интуитивной мотивацией для применения в данной ситуации обобщенных интервальных преобразований.

1. Обобщенные интервальные преобразования

В этой главе вводятся необходимые термины и понятия теории информации, которые потребуются нам в дальнейшем, дается определение обобщенных интервальных преобразований, кодирования с помощью обобщенных интервальных преобразований и приводятся результаты о стоимости кодирования с помощью этого метода с точки зрения теории кодирования источника.

Рассмотрим конечный вероятностный источник без памяти S (т.е. последовательность независимых случайных величин), порождающий буквы из конечного алфавита $\mathbf{A} = \{a_i\}$ с вероятностями p_i . Будем называть множество слов над алфавитом \mathbf{A} префиксным, если никакое слово из этого множества не является началом (префиксом) никакого другого. Рассмотрим также некоторое кодирование $f: \mathbf{A}^\infty \rightarrow \{0,1\}^\infty$, т.е. такое отображение из счетной прямой суммы алфавита во всевозможные последовательности нулей и единиц, что полный образ образует префиксное множество. Тем же символом f мы будем обозначать и ограничение $f: \mathbf{A}^N \rightarrow \{0,1\}^\infty$ исходного кодирования на все последовательности длины N над алфавитом \mathbf{A} , если это не приводит к неоднозначности. Обозначим через $|f(\mathbf{S})|$ длину последовательности из нулей и единиц для образа последовательности $\mathbf{S} \in \mathbf{A}^N$. Тогда стоимостью $C(f, S)$ кодирования будем называть предельное среднее число кодовых букв, приходящихся на одну входную:

$$C(f, S) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{\mathbf{S}, |\mathbf{S}|=N} p(\mathbf{S}) |f(\mathbf{S})|$$

Для любого подмножества $\mathbf{X} \subseteq \mathbf{A}$ будем обозначать через p_X вероятность этого множества, т.е.

$$p_X = \sum_{a_i \in X} p_i$$

Рассмотрим сообщение \mathbf{S} , составленное из букв алфавита $\mathbf{A} = \{a_i\}$, и разбиение алфавита \mathbf{A} на два подмножества \mathbf{A}_1 и \mathbf{A}_2 , т.е. $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$.

Определение 1. Частичным обобщенным интервальным преобразованием будем называть преобразование данного сообщения \mathbf{S} в три другие последовательности:

1) последовательность \mathbf{S}_1 , состоящую только из букв подмножества \mathbf{A}_1 , входящих в последовательность \mathbf{S} ; иначе говоря, \mathbf{S}_1 - это последовательность \mathbf{S} , из которой “вычеркнули” все буквы, входящие в \mathbf{A}_2 .

2) последовательность \mathbf{S}_2 , состоящую только из букв подмножества \mathbf{A}_2 , входящих в последовательность \mathbf{S} ; иначе говоря, \mathbf{S}_2 - это последовательность \mathbf{S} , из которой “вычеркнули” все буквы, входящие в \mathbf{A}_1 .

3) последовательность \mathbf{S}_I количества других букв (интервалов) между последовательными вхождением в \mathbf{S} любой буквы из подмножества \mathbf{A}_1 .

Подмножество \mathbf{A}_1 будем называть опорным подмножеством. Заметим, что в качестве опорного подмножества мы могли бы выбрать и подмножество \mathbf{A}_2 - тогда бы мы использовали интервалы между последовательными вхождением в \mathbf{S} любой буквы из подмножества \mathbf{A}_2 . Для иллюстрации определения рассмотрим на конкретном примере, как работает частичное интервальное преобразование. Пусть $\mathbf{A} = \{a, b, c\}$, $\mathbf{S} = aabcbccabbaca$, $\mathbf{A}_1 = \{a\}$ и $\mathbf{A}_2 = \{b, c\}$, то-

гда, используя обобщенное интервальное преобразование, получим три последовательности: $\mathbf{S}_1 = aaaaa$, $\mathbf{S}_2 = bcbccbbbc$ и $\mathbf{S}_I = 0,0,6,2,1$.

Пусть алфавит \mathbf{A} разбит на два подмножества \mathbf{A}_1 и \mathbf{A}_2 , т.е. $\mathbf{A} = \mathbf{A}_1 \cup \mathbf{A}_2$. С источником \mathbf{S} , порождающим буквы из алфавита \mathbf{A} , свяжем еще два источника: источник S_1 , порождающий буквы a_i из множества \mathbf{A}_1 с вероятностями $\frac{p_i}{p_{A_1}}$, и источник S_2 , порождающий буквы

a_i из множества \mathbf{A}_2 с вероятностями $\frac{p_i}{p_{A_2}}$. Обозначим через $S_{geom}(p)$ источник без памяти,

порождающий положительные целые числа n с вероятностями $p(1-p)^{n-1}$. Пусть также f_1 - некоторое кодирование источника S_1 , f_2 - некоторое кодирование источника S_2 и f_3 - некоторое посимвольное кодирование источника $S_{geom}(p)$. Заметим, что для стоимости кодирования f_3 источника $S_{geom}(p)$ будет верна несколько более простая формула, чем для общего случая:

$$C(f_3, S_{geom}(p)) = \sum_{n=1}^{\infty} p(1-p)^{n-1} |f_3(n)|$$

Определение 2. Будем называть f кодированием с помощью частичного обобщенного интервального преобразования (или просто частичным интервальным кодированием), если f отображает любое сообщение $\mathbf{S} \in \mathbf{A}^N$ в двоичные последовательности по формуле

$$f(\mathbf{S}) = l(\mathbf{S}_I) \parallel f_3(\mathbf{S}_I) \parallel f_1(\mathbf{S}_1) \parallel f_2(\mathbf{S}_2),$$

где \mathbf{S}_1 , \mathbf{S}_2 и \mathbf{S}_I - последовательности, полученные применением обобщенного интервального преобразования к \mathbf{S} по разбиению \mathbf{A} на подмножества \mathbf{A}_1 и \mathbf{A}_2 , $l(\mathbf{S}_I)$ - некоторый префиксный код длины последовательности \mathbf{S}_I , а \parallel - операция присоединения битовых последовательностей.

Очевидно, что слова $\{f(\mathbf{S}) | \mathbf{S} \in \mathbf{A}^N\}$ образуют префиксное множество и, следовательно, однозначно декодируемы. Можно пояснить однозначную декодируемость множества $\{f(\mathbf{S}) | \mathbf{S} \in \mathbf{A}^N\}$ иначе таким образом: сначала декодируем $l(\mathbf{S}_I)$ и знаем сколько символов надо декодировать из кода последовательности $f_3(\mathbf{S}_I)$ и $f_1(\mathbf{S}_1)$ (поскольку длины последовательностей \mathbf{S}_I и \mathbf{S}_1 всегда равны). Длину последовательности \mathbf{S}_2 вычисляем как разность суммы всех значений последовательностей \mathbf{S}_I и длины последовательности \mathbf{S}_I , поскольку сумма всех значений последовательности \mathbf{S}_I равна длине исходной последовательности \mathbf{S} . Приведем без доказательства лемму о связи стоимости кодирования с помощью частичного обобщенного интервального и стоимостями кодирования источников S_1 , S_2 и $S_{geom}(p)$.

Лемма 1.

Стоимость кодирования с помощью частичного обобщенного интервального преобразования вычисляется по формуле:

$$C(f, S) = p_{A_1} (C(f_1, S_1) + C(f_3, S_{geom}(p_{A_1}))) + p_{A_2} C(f_2, S_2)$$

Это утверждение интуитивно может быть обосновано с помощью следующих рассуждений: при достаточно большой длине N сообщения \mathbf{S} длина \mathbf{S}_1 будет асимптотически равна $p_{A_1}N$, длина \mathbf{S}_2 будет асимптотически равна $p_{A_2}N$, а длина последовательности \mathbf{S}_I равна длине последовательности \mathbf{S}_1 . И поскольку при росте N распределение вероятностей на интервалах будет стремиться к геометрическому распределению, становится понятно, что стоимости кодирования f_1 и f_3 войдут с весом p_{A_1} в итоговую стоимость кодирования, а стоимость f_2 войдет с весом p_{A_2} .

2. Модификация алгоритма JPEG LOCO-I

Для кодирования значение “ноль” ошибки предсказания отдельно от других значений необходимо, по существу, представить последовательность ошибок предсказания в виде двух последовательностей: первая – это последовательность интервалов между нулями в исходной последовательности, а вторая – это все ненулевые значения ошибки предсказания. Иначе говоря, необходимо применить к словам, порождаемым источником $S_{\hat{\varepsilon}}$ с алфавитом допустимых значений ошибки предсказания $A_{\hat{\varepsilon}} = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, n]\}$ и распределением вероятностей $P_{\rho}(\hat{\varepsilon})$, частичное интервальное преобразование с разбиением алфавита множествами $A_1 = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, 0) \cup (0, n]\}$ и $A_2 = \{0\}$.

На этом пути возникает ряд вопросов. Во-первых, в результате частичного интервального преобразования получаются три последовательности S_1 , S_2 и S_I , и их нужно далее как-то кодировать. Во-вторых, нам заранее не известны параметры распределения вероятностей $P_{\rho}(\hat{\varepsilon})$ значений ошибки предсказания.

Рассмотрим вопрос кодирования декомпозированных последовательностей ошибок предсказания, считая, что мы каким-то образом знаем параметры распределения. Заметим, что последовательность S_2 кодировать не нужно, поскольку она состоит из повторов одного символа (а именно, “0”). Относительно последовательности S_I мы знаем, что на последовательности интервалов (в данном случае между нулевыми значениями) имеется геометрическое распределение вероятностей. Поэтому, поскольку коды Райса-Голомба являются оптимальными для геометрически распределенных целых чисел, для кодирования последовательности S_I предлагается использовать коды Райса-Голомба. Последовательность S_1 – это все ненулевые ошибки предсказания. Отобразим множество $A_1 = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, 0) \cup (0, n]\}$ во множество $A'_1 = \{\hat{\varepsilon}_m \mid \hat{\varepsilon}_m \in [-n, n-1]\}$ следующим образом:

$$\hat{\varepsilon}_m = \begin{cases} \hat{\varepsilon}, & \hat{\varepsilon} < 0 \\ \hat{\varepsilon} - 1, & \hat{\varepsilon} > 0 \end{cases}$$

Для кодирования последовательности S_1 , записанной уже в алфавите A'_1 , мы делаем следующее эмпирическое предположение: будем считать, распределение вероятностей на множестве $A'_1 = \{\hat{\varepsilon}_m \mid \hat{\varepsilon}_m \in [-n, n-1]\}$ имеет также вид усеченной двухсторонней геометрической прогрессии $P_{\rho}(\hat{\varepsilon}_m)$, но со своими параметрами, отличающимися от параметров распределения для полного множества ошибок предсказания. Подтверждением оправданности применения этой гипотезы являются хорошие результаты степеней сжатия изображений, полученные при следовании этому предположению. Таким образом, для кодирования последовательности S_1 будет использоваться тот же метод кодирования, который применяется в JPEG LOCO-I, только подправленный на преобразование $\hat{\varepsilon} \rightarrow \hat{\varepsilon}_m$.

При сделанных предположениях относительно методов кодирования последовательностей S_1 и S_I теперь нужно определиться с методами оценки параметров распределений вероятностей на этих последовательностях и параметров кодов. Для оценки параметров распределения $P_{\rho}(\hat{\varepsilon}_m)$ на множестве $A'_2 = \{\hat{\varepsilon}_m \mid \hat{\varepsilon}_m \in [-n, n-1]\}$ и оценки параметров кода Райса-Голомба для кодирования S_1 будем применять метод, уже использованный в алгоритме JPEG LOCO-I.

Для оценки параметра кода Райса-Голомба при кодировании последовательности интервалов S_I применим следующий метод, предложенный в работе [4]. Выбирается некоторый разумный диапазон для значений параметра код Райса-Голомба. Для каждого такого значения хранится аккумулированная суммарная длина кода, как если бы именно это значение было выбрано для кодирования текущего значения интервала из последовательности S_I . А для кодирования следующего значения интервала используется то значение параметра код Райса-Голомба, у которого эта аккумулированная суммарная длина наименьшая. Такой метод, как показано в

работе [4], достаточно быстро сходится к истинному значению параметра кода Райса-Голомба для геометрического распределения вероятностей – и в теории, и на практике.

Будем называть вышеописанный алгоритм модифицированным JPEG LOCO-I и обозначать M-JPEG LOCO-I или M-JPEG LS. И прежде чем перейти к результатам практической реализации, посмотрим какие теоретические характеристики могут быть получены для стоимости кодирования и скорости адаптации при использовании предложенного метода.

3. Анализ алгоритма M-JPEG LOCO-I

Оценим теоретически, какую стоимость кодирования имеет предложенная методика кодирования и адаптации. Для этого введем следующие обозначения:

- f_L – кодирование, применяемое в JPEG LOCO-I;
- f_G – кодирование Райса-Голомба;
- f_M – предложенный метод кодирования с помощью частичного интервального преобразования;
- $S_{M,\hat{\epsilon}}$ – источник, порождающий буквы из алфавита $\mathbf{A}'_1 = \{\hat{\epsilon}_m \mid \hat{\epsilon}_m \in [-n, n-1]\}$ с вероятностями $P_\rho(\hat{\epsilon}_m)$;
- $S_{\hat{\epsilon}}$ – источник, порождающий буквы из алфавита $\mathbf{A}_{\hat{\epsilon}} = \{\hat{\epsilon} \mid \hat{\epsilon} \in [-n, n]\}$ с вероятностями $P(\hat{\epsilon})$ и связанный с источником $S_{M,\hat{\epsilon}}$, как указывалось ранее;
- $p_0 = P(0)$;
- N – длина последовательности ошибок предсказания $\mathbf{S}_{\hat{\epsilon}}$, порожденной источником $S_{\hat{\epsilon}}$.

Теорема 1.

1) В случае известного распределения вероятностей $P(\hat{\epsilon})$ источника $S_{\hat{\epsilon}}$ с алфавитом допустимых значений ошибки предсказания $\mathbf{A}_{\hat{\epsilon}} = \{\hat{\epsilon} \mid \hat{\epsilon} \in [-n, n]\}$ стоимость кодирования этого источника с помощью модифицированного метода f_M будет равна:

$$C(f_M, S_{\hat{\epsilon}}) = (1 - p_0)(C(f_G, S_{geom}(1 - p_0)) + C(f_L, S'_{\hat{\epsilon}})).$$

2) В случае неизвестных параметров распределения вероятностей $P(\hat{\epsilon})$ источника $S_{\hat{\epsilon}}$ при использовании методов адаптации, изложенных выше, средняя стоимость кодирования $\bar{C}(f_M, S_{\hat{\epsilon}}, N)$ будет отличаться от стоимости кодирования при известной статистике источника

не более чем на $O\left(\frac{1}{\sqrt{N}}\right)$:

$$\bar{C}(f_M, S_{\hat{\epsilon}}, N) = C(f_M, S_{\hat{\epsilon}}) + O\left(\frac{1}{\sqrt{N}}\right).$$

Доказательство:

Пункт 1) Теоремы сразу следует из Леммы 1, если в условиях Леммы 1 положить $f = f_M$, $f_3 = f_G$, $f_2 = f_L$.

Для доказательства пункта 2) приведем сначала результат из работы [4] о порядке скорости адаптации метода оценки параметра кода Райса-Голомба для геометрического распределения. Утверждение состоит в том, что для источника с фиксированным, но неизвестным распределением вероятностей, порождающего символы с геометрическим распределением, средняя стоимость кодирования последовательности длины N , порожденной таким источником, не превзойдет стоимость кодирования этого источника при известной статистике более чем

на $O\left(\frac{1}{\sqrt{N}}\right)$ бит:

$$\bar{C}(f_G, S_{geom}(p), N) = C(f_G, S_{geom}(p)) + O\left(\frac{1}{\sqrt{N}}\right).$$

Нам остается показать, что на последовательности интервалов на самом деле задано геометрическое распределение вероятностей. Отметим, что событие $\Theta =$ “буква из \mathbf{A}'_1 появилось в k -ый раз на n -ом месте в \mathbf{S} ” является рекуррентным [10], и поэтому имеет вероятностное распределение суммы k независимых одинаково распределенных случайных величин $T_{\alpha,1}$, причем все $T_{\alpha,1} = T_1$ имеют следующее вероятностное распределение:

$$P(T_1 = n) = f_n, \quad n = 0, 1, 2, \dots,$$

где f_n - вероятность того, что a_1 появилось в первый раз на n -ом месте в \mathbf{S} . В нашем случае, поскольку вероятность \mathbf{A}'_1 равна $1 - p_0$, поэтому $f_n = (1 - p_0)p_0^n$.

Таким образом, на последовательности интервалов возникает геометрическое распределение вероятностей, и при кодировании этой последовательности скорость адаптации метода М-JPEG LS не превзойдет $O\left(\frac{1}{\sqrt{N}}\right)$ бит.

Для завершения доказательства заметим, что при кодировании источника $S_{M,\varepsilon}$ ненулевых ошибок предсказания скорость адаптации метода LOCO-I не превосходит, как показано в [7,8], $O\left(\frac{1}{N}\right)$ бит. Таким образом, порядок скорости адаптации всего метода будет определяться только скоростью адаптации при кодировании последовательностей интервалов. Теорема доказана.

Завершая анализ алгоритма М-JPEG LS, обсудим некоторые дополнительные возможности, которые можно использовать для усовершенствования и дальнейшего развития этого метода. Когда вероятность значения “0” ошибки предсказания больше суммарной вероятности остальных значений ошибки, мы используем интервальное преобразование с разбиением алфавита множествами $\mathbf{A}_1 = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, 0) \cup (0, n]\}$ и $\mathbf{A}_2 = \{0\}$. Однако, в случае, когда вероятность значения “0” ошибки предсказания меньше суммарной вероятности остальных значений ошибки, то такое кодирование будет неэффективным, и нужно использовать другое частичное интервальное преобразование: с тем же разбиением алфавита $\mathbf{A}_\varepsilon = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, n]\}$, но теперь на множества $\mathbf{A}_1 = \{0\}$ и $\mathbf{A}_2 = \{\hat{\varepsilon} \mid \hat{\varepsilon} \in [-n, 0) \cup (0, n]\}$. Иначе говоря, для нового преобразования нужно выбирать в качестве опорного менее вероятное множество \mathbf{A}_1 для эффективного кодирования получающейся последовательности интервалов. То есть, можно к уже имеющейся методике оценки распределений вероятностей добавить способ указывать какое из множеств \mathbf{A}_1 или \mathbf{A}_2 нужно выбирать в качестве опорного. Для этого можно просто подсчитывать количество P_1 элементов из \mathbf{A}_1 в последовательности ошибок предсказания \mathbf{S}_ε и количество P_2 элементов из \mathbf{A}_2 в \mathbf{S}_ε , и, если $P_1 > P_2$, то опорным будет множество \mathbf{A}_2 , иначе опорным будет множество \mathbf{A}_1 .

Основанием для такого развития алгоритма М-JPEG LS в случае, когда статистика источника известна, может опять служить Лемма 1. Если в условиях Леммы 1 положить $f = f_M$ и $f_3 = f_G$ в случае, когда $p_0 < \frac{1}{2}$, $f_2 = f_L$, иначе $f_1 = f_L$. Тогда стоимость кодирования

$C(f_M, S_\varepsilon)$ модифицированного метода в случае, когда $p_0 < \frac{1}{2}$, будет

$$C(f_M, S_\varepsilon) = p_0 C(f_G, S_{geom}(p_0)) + (1 - p_0) C(f_L, S'_\varepsilon),$$

иначе

$$C(f_M, S_\varepsilon) = (1 - p_0) (C(f_G, S_{geom}(1 - p_0)) + C(f_L, S'_\varepsilon)).$$

4. Результаты работы M-JPEG LOCO-I

Теорема 1 дает нам некоторое представление относительно того, насколько эффективна наша модификация алгоритма M-JPEG LOCO-I. Однако, более важным с точки зрения практического применения является вопрос, как наш алгоритм работает на реальных данных – т.е. оценка того, насколько наши гипотезы относительно распределения вероятностей символов в искусственных и фотореалистичных изображениях соответствуют действительности. Для экспериментальной проверки M-JPEG LOCO-I был использован стандартный тестовый набор изображений Waterloo Repertoire GreySet1 (links.uwaterloo.ca/bragzone.base.html). Результат применения алгоритма M-JPEG LOCO-I приведен в таблице. Под скоростью кодирования мы понимаем количество бит, которым представляется в сжатом файле один байт (8 бит) несжатого изображения. Иначе говоря, если L_{src} - размер исходного файла, L_{coded} - размер сжатого файла, то

скорость кодирования – это $8 \frac{L_{coded}}{L_{src}}$. Для сравнения с алгоритмом M-JPEG LOCO-I были вы-

браны алгоритм JPEG LOCO-I и алгоритм еще одного стандарта JPEG 2000 [11]. Стандарт JPEG 2000 предназначен для сжатия полутоновых изображений без потерь качества и с потерями качества. Основой этого стандарта является вейвлет-преобразование и арифметическое кодирование; с точки зрения сложности, JPEG 2000 является более “тяжеловесным” алгоритмом, чем JPEG LS. Поскольку JPEG LS и M-JPEG LS являются алгоритмами сжатия изображений без потерь качества, то и JPEG 2000 запускался в режиме неискажающего кодирования.

Сравнительные результаты работы M-JPEG LS

Файл	Размер исходного файла	Размер файла, сжатого JPEG LS	Размер файла, сжатого M-JPEG LS	Размер файла, сжатого JPEG 2000	Скорость кодирования, JPEG LS	Скорость кодирования, M-JPEG LS	Скорость кодирования, JPEG 2000
Bird	65551	28436	29006	29580	3,47	3,54	3,61
bridge	65551	47435	47582	49096	5,789	5,807	5,992
camera	65551	35338	36117	37261	4,313	4,408	4,547
lena	65551	37530	37865	38789	4,58	4,621	4,734
goldhill	65551	43264	43340	45028	5,28	5,289	5,495
montage	65551	22307	22126	24276	2,722	2,7	2,963
horiz	65551	768	716	1827	0,094	0,087	0,223
circles	65551	1250	1187	7448	0,153	0,145	0,909
crosses	65551	3158	3295	8559	0,385	0,402	1,045
slope	65551	12872	10764	10857	1,571	1,314	1,325
squares	65551	632	589	1265	0,077	0,072	0,154
text	65551	13368	12397	34392	1,631	1,513	4,197
Среднее по фотореал.	65551	35718	36006	37338	4,35	4,39	4,56
Среднее по искусств.	65551	5341	4825	10725	0,65	0,59	1,31
Среднее по всем	65551	20530	20415	240315	2,51	2,49	2,93

В Табл. 1 занесены результаты работы всех вышеперечисленных алгоритмов. В верхней части Табл. 1 на сером фоне сгруппированы данные по фотореалистичным изображениям. Сразу ниже, на белом фоне, записаны данные по синтетическим изображениям. В нижней части таблицы приведены средние значения сначала по группе фотореалистичных изображений, затем по искусственным, и в самом конце – по всем файлам. Как можно видеть из приведенных в таблице данных, на фотореалистичных изображениях M-JPEG LS незначительно уступает JPEG LS на всех файлах, но превосходит JPEG 2000. При применении M-JPEG LS к искусственным изображениям этот алгоритм превосходит JPEG LS практически на всех файлах, и на всех таких изображениях M-JPEG LS и JPEG LS превосходят JPEG 2000. Сравнение этих трех алгоритмов очень наглядно видно на Рис.2 и Рис. 3. По входящим в набор Waterloo Repertoire GreySet1 искусственным изображениям модифицированный M-JPEG LS превзошел оригинальный

на 10%. На фотореалистичных изображениях новый алгоритм уступил (около 0.7%), однако по сумме всех файлов немного опередил JPEG LS - на 0.5%. JPEG 2000 уступил обоим алгоритмам M-JPEG LS и JPEG LS, причем на отдельных файлах (squares, crosses) более чем в два раза.

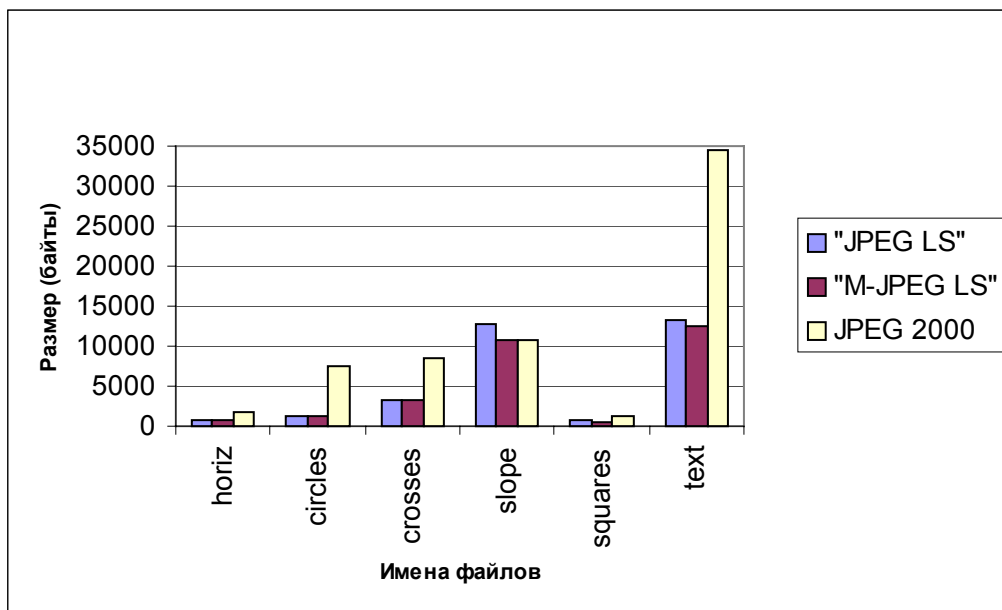


Рис. 2. Сравнение размеров искусственных изображений, сжатых разными алгоритмами

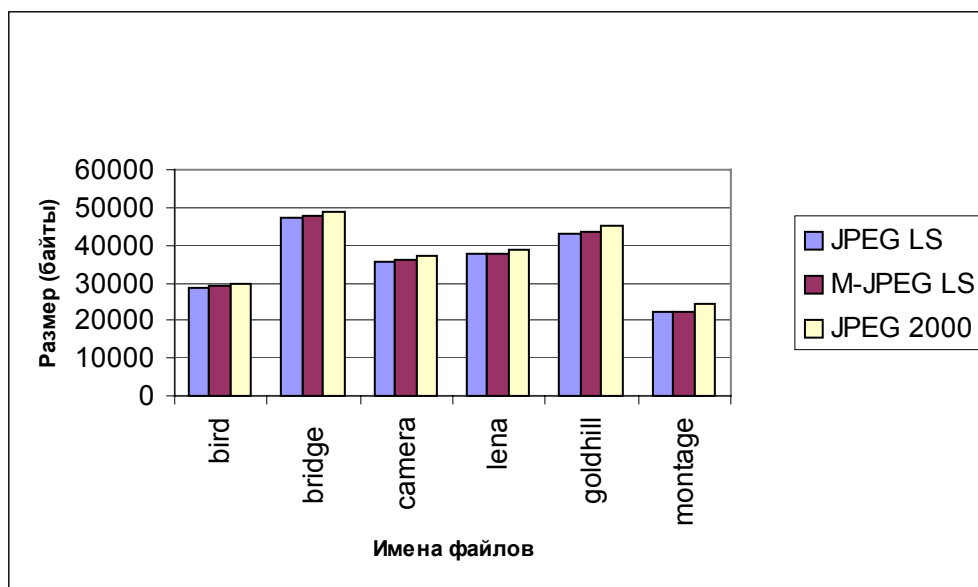


Рис. 3. Сравнение размеров фотореалистичных изображений, сжатых разными алгоритмами

Литература

1. ISO/IEC 14495-1,2, JPEG-LS Lossless and near-lossless compression of continuous-tone still images. - IS, 1999.
2. Todd S., Landgdon, G. G., Jr., and Rissanen J. Parameter reduction and context selection for compression of the gray-scale images // IBM Jr. Res. Develop., March 1985, vol. 29 (2). - P.188-193.
3. ISO/IEC 11544, Coded representation of picture and audio information – Progressive bi-level image compression. - IS, 1993.

4. Howard P.G., Witter J.S. Fast and efficient lossless image compression// Proc. IEEE Data Compression Conference, Snowbird, Utah, 1993. - P. 351-360.
5. Howard P.G., Witter J.S. Error modeling for hierarchical lossless image compression // Proc. IEEE Data Compression Conference, Snowbird, Utah, 1992. - P.269-278.
6. ISO/IEC 10918-1,2,3, JPEG (Digital compression and coding of continuous-tone still images). - IS, 1994,1995,1997.
7. Weinberger M.J., Seroussi G., Sapiro G. The LOCO-I Lossless Image Compression Algorithm: Principles and Standardization into JPEG-LS // IEEE Transactions on Image Processing, 2000, Vol.9.
8. Weinberger M.J., G. Seroussi On adaptive strategies for an extended family of Golomb-type codes // Proc. Data Compression Conference, 1997. - P. 131-140.
9. Netravali A., Limb J.O. Picture Coding: A Review // Proc. IEEE, 1980, Vol.68. - P.366-406.
10. Феллер В. Введение в теорию вероятностей и ее приложения, том 1: Пер. с англ. - М.: Мир, 1967.
11. ISO/IEC 15444-1, JPEG2000 JPEG 2000 image coding system. - IS, 2000.