РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 6 1994

УДК 681.3.053

В. О. Криворучко
(Новосибирск)

ОБ ОДНОМ МЕТОДЕ СЖАТИЯ ПОЛНОЦВЕТНЫХ СИНТЕЗИРОВАННЫХ ИЗОБРАЖЕНИЙ

Рассматриваются возможности для оптимизации предложенного ранее метода сжатия полноцветных синтезированных изображений. Приведены экспериментальные статистические данные работы алгоритма сжатия из класса LZW на полноцветных синтезированных изображениях. Предложен новый алгоритм сжатия на основе комбинации методов сжатия LZW и Хаффмана при абсолютной точности воспроизведения.

Введение. Уменьшение избыточности изображений становится все более актуальной задачей ввиду широкого распространения компьютерных систем для подготовки и обработки визуальной информации. При этом объем получаемой и хранимой информации оказывается слишком большим даже для современных носителей. Например, размер кадра изображения при минимальном для телевидения разрешении 384 × 288 пикселов и 24 битах на пиксел составляет 331766 байт. В последнее время наметилась тенденция к использованию изображений вплоть до 768 × 576 пикселов в кадре и при 32 и более битах на пиксел. Предложен улучшенный с точки зрения эффективности кодирования (сжатия) алгоритм на основе рассмотренного ранее в [1].

Алгоритм LZW. Напомним суть предлагавшегося ранее способа. Кадр изображения рассматривается как непрерывный поток данных, например, в соответствии с ходом телевизионной развертки. От некоторого текущего положения данные просматриваются «обратно» и находится максимально длинная последовательность пикселов, повторяющая ту, которая начинается с текущего пиксела. Найденная последовательность, или «цепочка», заменяется в выходном потоке на смещение ее местоположения в предыдущей части кадра относительно текущего пиксела и ее длину, или «счетчик», которые вместе образуют ссылку. После этого указатель текущего положения увеличивается на длину цепочки. Если соответствие между текущим и предыдущими пикселами не найдено, то текущий пиксел передается в выходной поток. Рассмотрим пример на рис. 1.

Входной поток:

1 2 3 4 5 6 7 8 2 3 7 2 3 4 9 2 3 4 6 ...

Выходной поток:

1 2 3 4 5 6 7 8 (−7, 2) 7 (−10, 3) 9 (−4, 3) 6 ...

Рис. 1

В выходной поток вводится один бит признака на каждый элемент, показывающий тип следующих за ним данных: пиксел или ссылку. И указатель, и длина цепочки не могут превышать некоторых величин, определяемых отведенными для их кодирования числом бит (разрядностью). Разрядность указателя задает окно просмотра. Данный метод кодирования обычно называют кодированием с кольцевым буфером.

Предложенный способ имеет ряд несомненных достоинств. Как было показано в [1], алгоритм обладает неплохими показателями коэффициента сжатия, отличается максимально простой реализацией декомпрессионной части и, что самое главное, легко настраивается на любую разрядность входных данных, не вызывая потерь информации при кодировании. Однако этот алгоритм в том виде, как он описан, на современных изображениях работает значительно хуже.

Тонкая настройка алгоритма. Алгоритм предусматривает наличие трех независимых параметров, оказывающих непосредственное влияние на коэффициент сжатия.

Одним из них является разрядность счетчика длины цепочки. При кодировании есть два противоположных фактора. С одной стороны, чем больше разрядность, тем более длинная цепочка может быть закодирована одной ссылкой. С другой стороны, коротких цепочек большинство, а значит, для их кодирования большая разрядность не нужна.

Другим параметром является разрядность указателя. Здесь похожая ситуация. Для того чтобы не пропустить достаточно длинную цепочку, необходимо просматривать пикселы в большом окне. С другой стороны, наиболее часто совпадения встречаются в пределах ближайших двух-трех строк и кодирование такого указателя с большой разрядностью неэффективно.

Важным параметром оказалась минимальная длина цепочки. При разрядности пикселов 24 бита и разрядности ссылки (указатель плюс счетчик) 16—18 бит длина цепочки, равная единице, имеет смысл, поскольку все равно есть сжатие. При меньшей разрядности пикселов необходимо увеличить минимальную длину цепочки до двух или трех.

Например, при разрядности счетчика 6 бит и минимальной длине цепочки, равной 1, диапазон длин 1—64 отображается в диапазон 0—63 и может быть закодирован 6 битами. В этом случае максимально достижимый коэффициент сжатия при 24 битах на пиксел и размере окна 2048 пикселов (т. е. для кодирования смещения используется 11 бит) можно найти как

$$64 \times 24/(11 + 6) = 90{,}4.$$

Изменяя разрядность счетчика в пределах от 4 до 8 бит и размер окна от 1024 до 4096 пикселов, можно выбрать оптимальные параметры для конкретного кадра.

Увеличение коэффициента сжатия. Исследования показывают, что возможности для оптимизации в этом направлении невелики. Однако есть еще одна возможность повысить коэффициент сжатия, состоящая в переходе к коду переменной длины.

Предложенное кодирование преобразует поток пикселов исходного кадра в поток данных, где есть три их типа: исходные пикселы, для которых не нашлось соответствия в пределах окна, счетчики длин цепочек и смещения цепочек. В исходном алгоритме эти данные кодируются без учета их статистики. Рассмотрим статистику этих типов данных.

На рис. 2 представлены типичные распределения числа найденных цепочек в зависимости от их длины для случая, когда счетчик кодируется 6 битами, на рис. 3 — в зависимости от их смещения относительно текущего пиксела для случая, когда смещение кодируется 11 битами.

Как видно из рис. 2, наиболее короткие цепочки встречаются наиболее часто и их можно было бы кодировать меньшим числом бит. Из рис. 3 видно, что чаще всего совпадения находились на расстоянии, равном примерно длине строки изображения или кратном ему, и такие указатели также должны быть закодированы меньшим числом бит. Наличие максимума в области смещений,

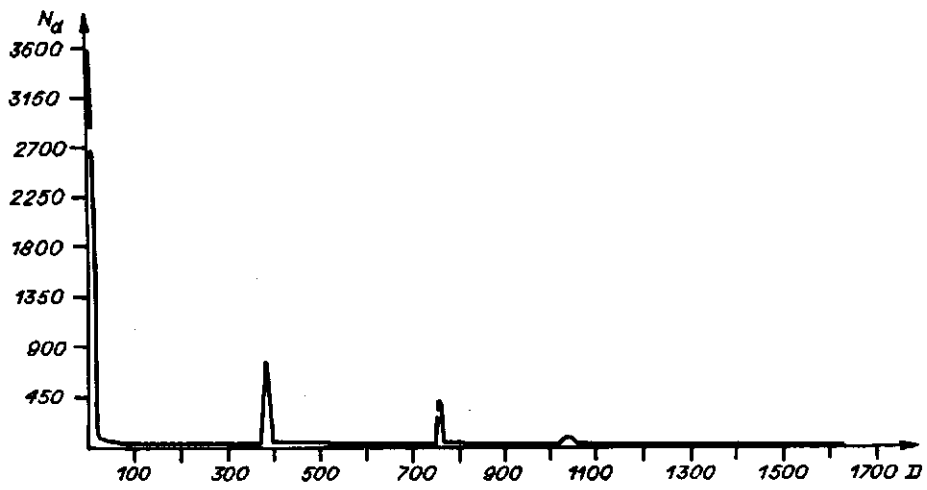

Рис. 2

равных единице, объясняется наличием большого числа цепочек одинаковых пикселов (серий). В исходный алгоритм выбора наиболее длинной строки необходимо внести некоторое дополнение, которое заключается в том, что если для данной цепочки пикселов находим две одинаковые соответствующие цепочки, то берем ту, которая располагается ближе к текущей. На рис. 1 для цепочки «2 3 4» есть два соответствия: со смещением −4 и −14. Необходимо взять цепочку со смещением −4, что и показано на рис. 1. Эта модификация не влияет на эффективность LZW-кодировки, однако создает более острое распределение по смещениям.

Теперь, когда статистика этих данных известна, можно построить эффективный код Хаффмана для дополнительного кодирования этих данных.

Часть пикселов всегда передается в выходной поток без изменения ввиду того, что для них не нашлось соответствия в пределах текущего окна. При усложнении исходного кадра число таких пикселов быстро растет и может достигать 50 %. Известно, что большинство соседних пикселов мало отличаются друг от друга. Таким образом, если рассмотреть, например, покомпонентную разность между текущим и предыдущим пикселами, то окажется, что

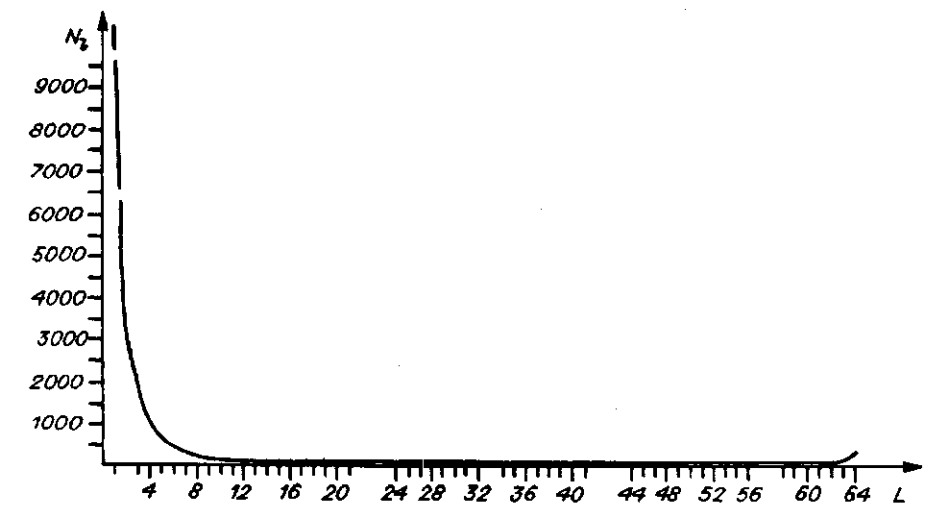

Рис. 3

Рис. 4

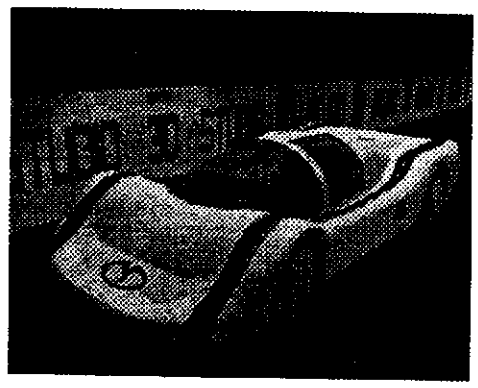

Рис. 5

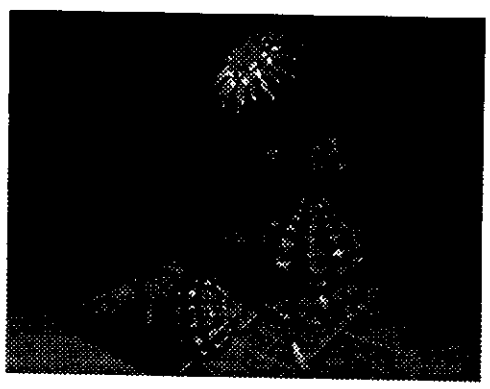

Рис. 6

Рис. 7

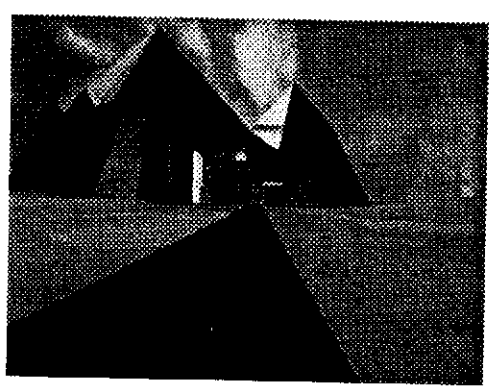

Рис. 8

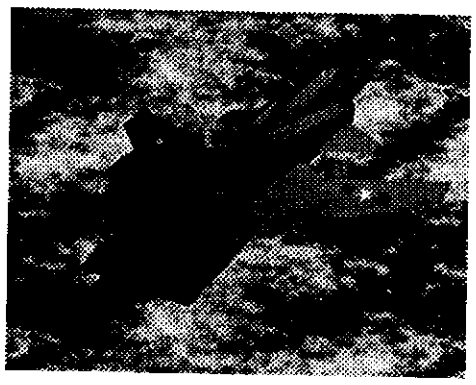

Рис. 9

чаще всего эта разность будет равна 0, +1, −1 в каждой из компонент, т. е. можно использовать кодирование с простейшим линейным предсказанием [2]. Применение более совершенного предсказания, чем просто разность, в данном случае затруднено наличием LZW-кодирования. Кодирование кодом Хаффмана разностей дает дополнительное сжатие.

Таким образом, исходный алгоритм преобразуется в двухпроходной. На первом проходе кадр кодируется по методу LZW. При этом если пиксел должен быть передан в выходной поток, записывается его покомпонентная разность с предыдущим пикселом. Параллельно набирается статистика по всем трем типам данных.

На втором проходе строится три таблицы кодов Хаффмана отдельно для счетчиков, смещений и разностей пикселов. Затем полученный на первом проходе файл перекодируется с использованием построенных кодовых таблиц. Таблицы кодов в этом случае должны быть сохранены вместе с данными для последующего декодирования.

Оказалось, что для разных изображений вид распределений отличается очень незначительно, что открывает возможность использования одних и тех же таблиц для всех изображений. В этом случае кодовые таблицы можно хранить отдельно от данных, что упрощает аппаратную реализацию алгоритма. Естественно, что для изображений с другим числом пикселов в строке максимумы распределений будут расположены в других местах и таблицы придется брать другие. С учетом этих замечаний был построен код Хаффмана, где статистика собиралась по всем тестовым изображениям.

Экспериментальные данные. Для экспериментов было использовано шесть кадров, синтезированных при помощи программного пакета 3D Studio v2.0 фирмы "Autodesk" с использованием фототекстур, закраски по методу Фонга и антиэлайзинга. Тестовые изображения представлены на рис. 4—9. Все изображения имеют разрешение 384 × 288 пикселов при 24 битах на пиксел. Результаты исследований представлены в таблице в виде номера рисунка, метода сжатия и соответствующего коэффициента сжатия.

В колонке LZW представлены результаты для алгоритма, предложенного в [1], т. е. просто LZW-кодирование. В колонке LZWHA и LZWHB приведены коэффициенты сжатия при использовании предложенного алгоритма, при этом применялись либо индивидуальные таблицы кодов Хаффмана (LZWHA), либо общая для всех тестовых изображений таблица кодов (LZWHB). В первом случае коэффициент сжатия вычислялся для кадра вместе с кодовой таблицей, а во втором — без таблицы, поскольку она хранится отдельно. В колонке TIFF показаны коэффициенты сжатия по методу LZW, предусмотренному стандартом TIFF [3], в колонке JPEG — коэффициенты сжатия тех же кадров по методу JPEG при отсутствии квантования, т. е. параметр качества устанавливался равным 100 %. Для первых трех методов автором были написаны соответствующие программы, для остальных — использовалась широко известная программа для обработки изображений ALCHEMY v1.4.

При кодировании методами LZW, LZWA, LZWB производился подбор параметров для разрядностей счетчиков и смещений в пределах: для счетчиков — от 4 до 8 бит, для смещений — от 10 до 12 бит. В таблице показаны наибольшие из полученных коэффициентов сжатия.

Из таблицы видно, что метод JPEG значительно превосходит по степени сжатия все остальные, поскольку все тестовые изображения синтезированы с использованием мощных антиэлайзинговых возможностей пакета 3D Studio и практически не содержат высоких гармоник. Напомним лишь, что JPEG в отличие от остальных проверяемых алгоритмов не обеспечивает побитовой точности восстановления изображений при декомпрессии. Напомним также, что в стандарте JPEG предусмотрена лишь работа с изображениями, где цветовое разрешение не превышает 24 бита на пиксел, т. е. максимум может быть три канала R, G и B при 8 битах на компоненту и, возможно, байтовый α-канал.

Рисунок	LZW	LZWHA	LZWHB	TIFF	JPEG
4	2,95	3,77	4,0	2,30	11,56
5	1,69	2,20	2,27	1,38	6,71
6	3,25	4,37	4,49	2,68	12,36
7	6,05	8,08	8,42	3,44	13,43
8	2,86	4,37	4,46	2,09	13,87
9	3,44	4,05	4,31	2,16	6,62

Предлагаемый алгоритм не накладывает никаких условий на цветовое разрешение изображений.

При сжатии с использованием предлагаемого алгоритма и индивидуальных кодовых таблиц (LZWHA) эффект оптимального для конкретного кадра кодирования ослабляется тем, что необходимо хранить вместе со сжатым кадром три кодовых таблицы, при этом их суммарный размер может достигать 2000 байт для каждого из кадров. Поэтому метод LZWHB оказался предпочтительнее на всех тестовых кадрах.

Заключение. Предлагаемый алгоритм комбинированного LZW — Huffman сжатия полноцветных изображений можно рекомендовать в качестве одного из стандартных методов сжатия полноцветных изображений там, где требуется побитовая точность воспроизведения, либо там, где нежелательно использование JPEG, например, ввиду его сложности и ограничений, накладываемых на число бит в пикселе. Если предполагается работать лишь с одним форматом изображений, то можно создать некоторый стандартный набор кодовых таблиц Хаффмана с тем, чтобы не хранить их вместе со сжатым изображением. Если формат изображения отличается от стандартного, то рекомендуется использовать алгоритм LZWHA. Использование предлагаемого алгоритма вместо стандартного TIFF LZW сжатия обеспечивает выигрыш по степени сжатия в среднем в 2 раза.

СПИСОК ЛИТЕРАТУРЫ

1. **Криворучко В. О.** Сжатие цветных синтезированных изображений // Автометрия.— 1993.—№ 5.
2. **Ефимов В. М., Золотухин Ю. Н., Колесников А. Н.** Оценка эффективности некоторых алгоритмов сокращения избыточности информации при абсолютной точности воспроизведения // Автометрия.—1991.—№ 6.
3. **Романов В. Ю.** Популярные форматы файлов для хранения графических изображений.— М.: Унитех, 1992.

Поступила в редакцию 27 декабря 1993 г.