

## Повышение эффективности сжатия компьютерной графики с использованием RLE алгоритма

*А.С. Карпенко, И.В. Крысова*

*Омский государственный технический университет, Омск*

**Аннотация:** На скорость работы веб-ресурса в значительной степени влияют подгружаемые внешние объекты. Наиболее часто используемые из них – изображения. В связи с этим становится актуальным уменьшение размера подгружаемой компьютерной графики (изображений), чтобы уменьшить общий размер веб-страницы. Среди алгоритмов сжатия наименее востребованы алгоритмы кодирования без потерь, по причине их малой эффективности. Однако для определенного класса изображений, за счёт использования его специфических особенностей, можно получить значительно больший коэффициент сжатия при кодировании без потерь. Авторами предложена модификация RLE алгоритма, позволяющая увеличить коэффициент сжатия изображений с большими одноцветными областями.

**Ключевые слова:** RLE алгоритм сжатия, коэффициент сжатия, обработка данных, оценка эффективности алгоритма сжатия, длина последовательности.

Повышение эффективности сжатия компьютерной графики является на сегодняшний день актуальной задачей не только для современных веб-ресурсов. Это связано с большой распространённостью цифровых изображений в различных предметных областях и чрезвычайно большим ростом размеров цифровых изображений. В связи с этим решающее значение приобретает особый вид обработки изображений – их кодирование с целью сокращения объёма (компрессии) данных.

В данной работе рассмотрен наиболее востребованный RLE алгоритм сжатия без потерь, так же предложена его модификация, способная увеличить коэффициент сжатия для определенного класса изображений, в сравнении с классической реализацией.

Целью данной работы является математическое обоснование эффективности предложенной авторами модификации RLE алгоритма сжатия.

RLE алгоритм сжатия (Run Length Encoding – кодирование серий последовательностей) является наиболее известным и простым алгоритмом

сжатия информации обратимым путем [1-4]. Суть данного подхода состоит в замене цепочек, серий повторяющихся байтов или их последовательностей на один кодирующий байт и счетчик числа их повторений. Первый байт указывает сколько раз нужно повторить следующий байт. К положительным сторонам алгоритма можно отнести то, что он не требует дополнительной памяти при работе, и быстро выполняется. Данный метод, как правило, достаточно эффективен для сжатия растровых графических изображений (BMP, PCX, TIFF), т.к. последние содержат достаточно длинные серии повторяющихся последовательностей байтов. Недостатком метода RLE является достаточно низкая степень сжатия или стоимость кодирования файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях [5-7].

В классической реализации алгоритма RLE последовательность одинаковых элементов заменяется парой «служебный байт + повторяющийся элемент». Старший бит служебного байта отвечает за тип последовательности (одиночный элемент - 0, либо серия - 1), а оставшиеся семь бит выделяются под длину последовательности. Таким образом, максимальная длина кодируемой последовательности  $k$  может составлять не более 128 символов ( $k=2^7$ ) [8].

[Служебный байт] = 1 0 0 0 1 0 0

Данный алгоритм осуществляет сжатие без потерь, поэтому наиболее информативным критерием оценки эффективности работы алгоритма будем считать коэффициент сжатия  $R$  [9].

$$R = \frac{L_{исх}}{L_{сж}}$$

где  $L_{исх}$  – количество исходной информации;  $L_{сж}$  – количество сжатой информации.

Для наглядности приведен идеализированный пример, когда изображение представляет собой большую одноцветную область  $M_1$



одинаковых элементов  $a$ . Классический RLE алгоритм будет заменять каждые 128 элементов парой «служебный байт  $s_i$  + повторяющийся элемент  $a_i$ »:

$$M_1 = \{a_1, a_2, \dots, a_{100}, \dots, a_{1000}\} = 1000 \text{ байтов,}$$

$$M_2 = \{s_1, a, s_2, a, \dots, s_8, a\} = 16 \text{ байтов,}$$

где  $M_1$  – исходная последовательность;  $M_2$  – сжатая последовательность;  $s_n$  – служебный байт;  $a$  – повторяющийся элемент.

Коэффициент сжатия  $R$  для этого примера составит **62,5**.

В случае сжатия изображений с большими одноцветными областями эффективно резервировать большой объем памяти под длину цепочки [10]. В этих целях авторами предлагается выделить в служебном байте один бит под индикатор длинной цепочки и добавить дополнительный байт, содержащий в себе информацию о длине последовательности одинаковых символов:

$$[\text{Служебный байт 1}] = \mathbf{11}00100$$

$$[\text{Служебный байт 2}] = 1000100$$

Старший бит «служебного байта 1» отвечает за тип последовательности (одиночный элемент - 0, либо серия - 1), второй бит является индикатором длинной цепочки (последовательность менее 64 элементов – 0, а более 64 элементов – 1), а оставшиеся шесть бит выделяются под длину последовательности. «Служебный байт 2» полностью содержит информацию о длине последовательности одинаковых символов.

В итоге максимальная длина кодируемой последовательности  $k$  может составлять 320 символов ( $k=2^6+2^8$ ). Что в два с половиной раза больше классической реализации RLE алгоритма.

Рассмотрим вышеописанный пример большой одноцветной области одинаковых элементов  $M_1$ :

$$M_1 = \{a_1, a_2, \dots, a_{100}, \dots, a_{1000}\} = 1000 \text{ байтов}$$

Модифицированный RLE алгоритм будет заменять каждые 320 элементов тройкой «служебный байт 1  $s_i$  + служебный байт 2  $s_i$  + повторяющийся элемент  $a_i$ »:

$$M_2 = \{s_1, s_2, a, s_3, s_4, a, s_5, s_6, a\} = 9 \text{ байтов}$$

При этом коэффициент сжатия составит  $R=111,1$ .

Таким образом, за счёт увеличения количества элементов в каждой кодируемой цепочке возможно почти в два раза уменьшить количество сжатой информации и соответственно увеличить коэффициент сжатия  $R$ .

Важно отметить, что алгоритм наиболее эффективен для изображений деловой графики (чертежи, схемы и другие изображения с большими одноцветными областями), т.к. недостатком RLE алгоритма является достаточно низкая степень сжатия файлов с малым числом серий и, что еще хуже - с малым числом повторяющихся байтов в сериях.

### Литература

- [1] Тропченко А.Ю., Тропченко А.А. Методы сжатия изображений, аудиосигналов и видео. – СПб: СПбГУ ИТМО, 2009. – 108 с.
- [2] Bell, T. C., Moffat, A., Witten, I. H. Compressing the Digital Library // Proc. Digital Libraries '94. - Texas: College Station, 1994. - С. 41-46.
- [3] Jiawan Zh., Jizhou S., Zhigang S. Accelerate volume splatting by using run length encoding // Lecture Notes in Computer Science. - 2003. - Т. 2657. - С. 907-914.
- [4] Янишевская А.Г., Чурсин М.А. Способы хранения и обработки большого объема данных с использованием MapReduce и Persona Server // Инженерный вестник Дона. - 2015. - №2 ч .2. - С. 55. URL: <http://www.ivdon.ru/ru/magazine/archive/n2p2y2015/2976>.
- [5] Бугаенко П.С., Пасечник С.В., Янишевская А.Г. Использование метода фрактального сжатия // Информационные технологии и технический дизайн в профессиональном образовании и промышленности сборник материалов V Всероссийской научно-практической конференции с международным участием. - Новосибирск: НГТУ, 2013. - С. 74-78.



[6] Карпенко А.С., Крысова И.В. Использование графического формата MNG для сжатия изображений в веб-программировании. – Информационные технологии в науке и производстве : материалы молодежной науч.-техн. конф.– Омск : Изд-во ОмГТУ, 2015. С. 91-94.

[7] Клаус А.И. Редактор изображений с использованием технологии флеш как элемент веб-приложения: проблемы и решения // Инженерный вестник Дона. - 2009. - №2. - С. 59. URL: <http://www.ivdon.ru/ru/magazine/archive/n2y2009/136>

[8] Алгоритмы сжатия RLE и LZ77 // Habrahabr.ru: коллективный блог URL: <http://habrahabr.ru/post/141827/> (дата обращения: 21.11.2015).

[9] Ватолин Д., Ратушняк А., Смирнов М., Южин В. Методы сжатия данных. Устройство архиваторов, сжатие изображений и видео. – М.: ДИАЛОГ-МИФИ, 2002 – 384 с.

[10] Карпенко А.С., Крысова И.В. Использование модифицированного RLE-алгоритма для сжатия графической информации // Россия молодая: передовые технологии - в промышленность. - 2015. - №3. - С. 16-20.

### References

[1] Tropchenko A.Ju., Tropchenko A.A. Metody szhatija izobrazhenij, audiosignalov i video [Methods of image compression, audio and video]. – SPb: SPbSU ITMO, 2009. 108 p.

[2] Bell, T. C., Moffat, A., Witten, I. H. Compressing the Digital Library. Proc. Digital Libraries '94. - Texas: College Station, 1994. С. 41-46.

[3] Jiawan Zh., Jizhou S., Zhigang S. Accelerate volume splatting by using runs length encoding. Lecture Notes in Computer Science. 2003. Т. 2657. С. 907-914.

[4] Yanishevskaya A.G., Chursin M.A. Inzhenernyj vestnik Dona (Rus). 2015. №2 p.2. pp. 55. URL: [ivdon.ru/ru/magazine /archive/n2p2y2015/2976](http://ivdon.ru/ru/magazine/archive/n2p2y2015/2976).



[5] Bugaenko P.S., Pasechnik S.V., Yanishevskaya A.G. Informacionnye tehnologii i tehniceskij dizajn v professional'nom obrazovanii i promyshlennosti sbornik materialov V Vserossijskoj nauchno-prakticheskoy konferencii s mezhdunarodnym uchastiem. Novosibirsk: NSTU, 2013. pp. 74-78.

[6] Karpenko A.S., Krysova I.V. Informacionnye tehnologii v nauke i proizvodstve : materialy molodezhnoj nauch.-tehn. konf. Omsk : OmSTU, 2015. pp. 91-94.

[7] Klaus A.I. Inzhenernyj vestnik Dona (Rus). 2009. №2. pp. 59. URL: [ivdon.ru/ru/magazine/archive/n2y2009/136](http://ivdon.ru/ru/magazine/archive/n2y2009/136)

[8] RLE compression algorithms and LZ77. URL: [habrahabr.ru/post/141827/](http://habrahabr.ru/post/141827/).

[9] Vatolin D., Ratushnyak A., Smirnov M., Juzhin V. Metody szhatija dannyh. Ustrojstvo arhivatorov, szhatie izobrazhenij i video [Data compression methods. The device archives, image and video compression]. M.: DIALOG-MIFI, 2002. 384 p.

[10] Karpenko A.S., Krysova I.V. Rossiya molodaja: peredovye tehnologii - v promyshlennost'. 2015. №3. pp. 16-20.