computing@computingonline.net
www.computingonline.net

ISSN 1727-6209
International Journal of Computing

A CHOICE OF THE SHORTEST OF THE ALTERNATIVE COMPRESSED BLOCKS OF THE DYNAMIC HUFFMAN'S CODES IN THE FORMAT OF PNG

Alexander Shport'ko

Department of Informatics and Applied Mathematics
Rivne State Humanitarian University, Ukraine
33028, Rivne, 12, S. Bandery street
E-mail:chportko@ukr.net, chportko@yandex.ru

Resume: *in the article the algorithm of generation of the alternative compressed blocks of dynamic Huffman's codes for every block of data, choice of the shortest compressed block of alternative and iterative diminishing of its size for the improvement of the compression of images in the format of PNG is offered. The methods of the estimation of the size of block of Huffman's codes according to the absolute frequencies of its elements are examined in details. The fragment of programs outlaying the language of C, which allows exactly to determine the size of the compressed block by principle of generation of dynamic Huffman's codes and separate working of small frequencies of elements, using for this purpose on the average not more time, than for the close estimation of size with the use of entroping. As experiments show, realization of the offered algorithm, allows to improve the indexes of the compression of the majority of images in the format of PNG in 2 – 6 %.*

Keywords*: lossless compression of images, dynamic Huffman's codes, format of graphic files of PNG.*

INTRODUCTION

The format of graphic files of PNG was created on October, 1, 1996 for the effective preserving of bitmapped images without losses after the company "Unisys" began to require paying for the using of format of GIF [1]. In this work the method of increase of indexes of compression of images in this format due to the generation and comparison of sizes of the alternative compressed blocks is described in details.

In PNG-files which are used nowadays, compressed data are saved in separate blocks according to the format of dictionary compression of Deflate [1, 3]. In accordance to this format, the compressed blocks contain results of he using of the input stream of predictors and algorithm of the LZH [3], according to which the results of context-dependent dictionary algorithm of the LZ77 [4] are compressed by the context-independent Huffman's codes.

Describing the dictionary algorithms, the fixed amount of the previous coded indivisible elements (literals) of the input stream is called a dictionary, but next uncoded – ones a buffer. The algorithm LZ77 is based on the replacement in the output stream of sequence of next literals of buffer by sending the similar sequence of literals of the dictionary as a pair of numbers *<length; displacement is from the end of the dictionary>*. In the case of absence of the similar sequence of literals in a dictionary, the first literal of buffer is transferred in the output stream began without changes. After this coded literals are transferred from the beginning of buffer to the end of the dictionary and coding continues in the same way up to the ending of literals of input stream. For the providing of synonymousness of decoding separate literals and base values of lengths are encoded together. With the purpose of increasing of indexes of compression for displacements and literals/lengths different optimum dynamic Huffman's codes are generated in every block of the compressed data.

1. ALGORITHM OF THE GENERATION OF THE ALTERNATIVE COMPRESSED BLOCKS, CHOICE OF THE SHORTEST BLOCK OF THE ALTERNATIVE AND ITERATIVE DIMINISHING OF ITS SIZE

Obviously, replacements of LZ77 are considered to be effective, if they are written down by not

greater amount of bats, than separate literals, which they replace. As separate literals and base values of lengths and displacements in the format of Deflate are written down by the codes of Huffman, the replacement of j length of len_j, which is implemented beginning with literal of s_k for displacement of $offset_j$ is effective only, when executed (3), where l_m is the length of Huffman's codes of literal/length of replacement of m, d_m is the quantity of the additional bats for the recording of length of replacement of m, λ_m – length of Huffman's code of displacement of m, δ_m is the quantity an amount of additional bats for the record of displacement of m. But short replacements appear, as a rule, effective for the compressed blocks in which other replacements of the same length are also taken into account in. It is connected with the fact that taking of replacements of identical length can substantially increase the frequency of this length of replacement in the distributing of literals/lengths and, as a result, decrease the length of its Huffman's codes. On the other side, taking into account of the aggregate of short replacements can substantially decrease frequencies of separate literals and as a result multiply lengths of their Huffman's codes. Besides, the displacement between the identical fragments of entry block of data substantially influence the length of the compressed blocks because greater displacements are encoded by greater quantity of additional bats.

According (3) to the standard of Deflate, for any distributing of frequencies there will be effective replacements not less than 48 literals. In practice, images in which the separate value of brightness repeats more frequent by than other values of brightness taken together, meet extraordinarily seldom (for example solid fillings), as a result – lengths of codes of literals almost always exceed 1, that is why effective are supposed replacements not less than 24 elements. The efficiency of shorter replacements depends not only on literals, which they replace, but also on the distribution of the compressed block in general. It is impossible to define the character of literals/length distribution in regular block and moreover, the distribution of displacement for every block. **That is why we offer the following algorithm of minimization of size in proper compressed block:**

1) **to create the alternative compressed blocks with different maximal lengths of not taken into account replacements;**
2) **to choose the shortest block among them;**
3) **decrease iteratively its length by casting aside of not effective and accounting of effective replacements according to (3);**
4) **to use the formed block for data storage in the format of Deflate PNG-file.**

2. RESULTS OF EXPERIMENTS

Let's examine the results of the application of the described algorithm of the use of the alternative compressed blocks for a compression, for example, of eight different type 24-bits images of standard set of ACT files of in the format of PNG. Testing was conducted with the help of the modified program from [1]. In fig. 1 image compressing factors are given, and in fig. 2 is the proper time of compression (s). Testing results of the program are given without application of the considered algorithm it is resulted in the third, and with application – in the fourth column. Besides, for the comparison of the efficiency of compression in the second column of tables the results of testing of "Microsoft Photo Editor 2000", which does not apply predictiors, are given and in the fifth and sixth there are results of the popular among Web-designers program OptiPng (http://www.optipng.sourceforge.net), which generates short PNG-files as a result of accordingly standard and maximal surplus of predictors, sizes of the compressed blocks and compression strategies.

As the results of testing show, application of the described algorithm increased an aspect ratio in 2 – 6% for 63% images, which, as a rule, are continuously toned, and did not influence the compression of the other files, although slowed implementation of the program on the average of 7%. Besides, the realization of the considered algorithm allowed to approximate to the aspect of the program of surplus ratio, for some images to surpass it, outlaying for a compression in 1.9 – 66 less time.

3. CONCLUSIONS

1. For the increase of the efficiency of data compression in formats which use the algorithms of a few methods consistently, we should take into account the mutual influence of these algorithms.
2. The considered algorithm substantially allows to improve compression factor of most images ratio due to choosing the shortest of the alternative compressed blocks of dynamic Huffman's codes and iterative diminishing of its size for every block.
3. The described algorithm does not require modification of decoder or programs of image viewing and due to decreasing the size of files only accelerates their work. For this reason, it can be effectively used for saving data in standards, which use the format of dictionary compression of Deflate.