Compressing High Bit Depth Images of Sparse Histograms

Roman Starosolski

Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

Abstract. To improve the lossless compression ratios for images having sparse histograms, a method of histogram packing was introduced. The method was found to be effective for low bit depth images. We investigate effects of packing histograms of high bit depth images—medical CR, CT, and MR images as well as various natural 16-bit ones. We analyze an off-line packing method, which requires encoding the original histogram along with the compressed image. We present several methods of histogram encoding and analyze their usefulness. One of them (RLE+LZ77) obtains the shortest encoded histogram length for nearly all tested images and in practice is sufficiently good for encoding histograms of wide range of images. A simpler method (MT) may be useful for medical images. For these images, its use results in improvements of the compression ratio little worse compared to RLE+LZ77, but decoding of images with histograms encoded using the MT method is already supported by JPEG-LS and JPEG2000 (part 2) standards. Effects of histogram packing are examined for the CALIC, JPEG2000, and JPEG-LS algorithms. Histogram packing improves significantly lossless compression ratios for high bit depth sparse histogram images. The ratio improvement may exceed a factor of two, as in the case of MR medical images.

Keywords: image coding, lossless image compression, high bit depth images, sparse histogram, histogram packing.
PACS: 07.05.Pj; 42.30.Va; 42.68.Sq; 95.75.Mn

INTRODUCTION

Most single-frame single-band medical images, like CR, CT, and MR, are of a high nominal bit depth, which usually varies from 12 to 16 bits per pixel. The number of active levels, i.e., intensity levels actually used by image pixels, may be smaller, than implied by the nominal bit depth, by an order of magnitude or even more. Furthermore, active levels are distributed throughout almost all the entire nominal intensity range, i.e., the images have sparse histograms of intensity levels. Also, the continuous tone natural (photographic) images of high bit depths may have sparse histograms due to acquisition device characteristics or to processing applied to images (like gamma correction or contrast adjustment). Histograms of some images are inherently sparse. Although this observation probably won't lead to improving the compression ratios for such images, we note the obvious fact: regardless of the nominal bit depth, the number of active levels cannot be greater, than the number of pixels. All in all, sparse histograms occur frequently in high bit depth images.

Image compression algorithms are based on sophisticated assumptions about images they process. Sparse histogram is clearly different from what is expected by most lossless image compression algorithms, both in the case of predictive and of transform coding. The impact of histogram sparseness on compression ratios of low bit depth sparse histogram images is well known—applying to such images a histogram packing may lead to significant ratio improvement [1,2]. An off-line histogram packing simply maps all the active levels to the lowest part of the nominal intensity range (order-preserving one-to-one mapping). The off-line packing requires the information, describing how to expand the histogram, to be encoded along with the compressed image; i.e., we have to encode the original histogram. There are also other methods targeted at sparse histogram images (for overview see [3]), however, the off-line packing has significant practical advantages over others. Several algorithms along with the compressed image store the "image palette" (PNG [4]) or the level "mapping table" (JPEG-LS [5]). For these algorithms, if we use off-line packing prior to compression, then the decompression reconstructs image and its original histogram solely by means of the algorithm (i.e., no additional step of histogram expanding is required after decompression). To our best knowledge, except for the first stage of the research reported herein [3], the compression of high bit depth images having sparse histograms has not been investigated. High bit depth images require the histogram to be encoded efficiently—in this paper we analyze methods of encoding histograms of high bit depth images as well as effects of histogram packing on compression ratio obtained using CALIC, JPEG2000, and JPEG-LS algorithms.

EFFICIENT ENCODING OF THE HISTOGRAM

The off-line histogram packing method actually is an image transform; we apply it to an image before the compression. It transforms sparse histogram image into the packed histogram image. The transform is reversible if, along with the compressed image, we encode the original histogram. For the histogram expanding, it is enough to encode which of intensity levels are active—we do not need to know how many times the active level was used.

In the case of 8-bit images, we may simply encode binary all the active levels. Following the JPEG-LS terminology, we call this method of histogram encoding the Mapping Table. For encoding a histogram of an N-bit image containing L active levels we need $(L + 1)\,N$ bits. For 16-bit images, in the worst case (all levels active), we'd need 128 kilobytes.

Instead of encoding the intensity level of each active level, we may encode, for all nominally available levels, the information whether the specific level is active. Therefore, we need 2^N bits to encode the histogram of an N-bit image, regardless of the number of active levels. This method of histogram encoding was used for 8-bit images in the EIDAC algorithm starting from its first version [1]. We call a histogram encoded in this manner the Bit-Array of the histogram. For a 16-bit image, the Bit-array requires 8 kilobytes; for an 8-bit image, 32 bytes only.

Some images, like MR images used for experiments in this paper, use below 1% of all the nominally possible levels. A histogram of such image, encoded using the Bit-Array method, contains long runs of 0's separated by single 1's. Such histogram could be represented more compactly if we encoded lengths of runs of 0's. If, on the other hand, the histogram is not sparse, then it contains long runs (or just one long run) of 1's. Therefore we encode the Bit-Array of the histogram using the Run Length Encoding (RLE) variant described in the Table 1. Encoding the histogram using the RLE method is most efficient when the number of levels is close to 0 or close to 2^N. In the worst case, i.e., when every second level is used, we need 2^{N+2} bits for the RLE encoded histogram—32 kilobytes for the worst case histogram of a 16-bit image.

TABLE 1. Run Length Encoding of histograms of images of bit depths up to 16 bits.

RLE codeword	Sequence
$0\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$	run of $r + 1$ 0's followed by single 1, $r = b_6 \ldots b_0$, $r < 126$
$0\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ b_7 \ldots b_0$	run of $r + 127$ 0's followed by single 1, $r = b_7 \ldots b_0$
$0\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ b_{15} \ldots b_0$	run of $r + 383$ 0's followed by single 1, $r = b_{15} \ldots b_0$
$1\ b_6\ b_5\ b_4\ b_3\ b_2\ b_1\ b_0$	run of $r + 1$ 1's followed by single 0, $r = b_6 \ldots b_0$, $r < 126$
$1\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ b_7 \ldots b_0$	run of $r + 127$ 1's followed by single 0, $r = b_7 \ldots b_0$
$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ b_{15} \ldots b_0$	run of $r + 383$ 1's followed by single 0, $r = b_{15} \ldots b_0$

The Bit-Array is inefficient when the number of active levels is low; the RLE may be inefficient for certain numbers of intensity levels. Fortunately, both the Bit-Array of the histogram and the RLE encoded histogram may be further compressed. In the cases, when the above methods are most inefficient, the histograms encoded using them are likely to contain multiple repetitions of long sequences of symbols (bits or RLE codewords). For compressing such data we may use a universal compression algorithm capable of capturing long contexts, like the LZ77 universal dictionary compression algorithm [6].

EXPERIMENTAL RESULTS

In order to evaluate the impact of histogram sparseness on compression ratio for typical medical image of a certain modality, we used all the CR, CT, and MR medical images from a test image set described in another study [7]. There were 12 images of each of the modalities; not all the medical images are of 16-bit depth and not every medical image has sparse histogram. Obviously, for the 10- or 12-bit images the method of histogram encoding gets less important for the overall compression ratio. Natural continuous tone grayscale images of 16-bit depth were included in experiments to evaluate effects of histogram packing on various non-medical images. These images included unprocessed images of various sizes as well as processed ones—for gamma and contrast adjustment we used Adobe Photoshop 9.0. Following groups of non-medical images were evaluated, each containing 4 images: natural (photographic) images of 16-bit depth classified in [7] as medium-sized (Medium), Medium images with contrast increased by 25% (Contrast), Medium images with gamma (value 1.25) correction applied (Gamma), and small images containing below 2^{16} pixels, which are reduced size Medium images (Small).

The characteristics of images and the results of encoding histograms are reported in the Table 2 (for brevity we report averaged results only). To characterize numerically image sparseness, we define the image level utilization $U = L \,/\, (1 + l_{hi} - l_{lo})$, where l_{lo} and l_{hi} are respectively the lowest and the highest active level, and L is number of active levels. In the tables, images are characterized by the image name, size (number of pixels), nominal depth (N), nominal (2^N) and actual (L) number of intensity levels, and by the level utilization (U). Sizes (in bytes) of encoded histograms are reported in the Table 2 for the following methods: Mapping Table (MT), Bit-Array (BA), Bit-Array compressed using LZ77 (BA+LZ77), RLE, and RLE method followed by LZ77 (RLE+LZ77); for the LZ77 compression we used the gzip compression utility (version 1.2.4).

TABLE 2. Comparison of histogram encoding methods (averages for groups).

Images						Encoded histogram size [B]				
Name	**Pixels**	***N***	**2^N**	***L***	***U***	**MT**	**BA**	**BA+ LZ77**	**RLE**	**RLE+ LZ77**
CR	3527076	12.5	23296	7878	59.5%	15184	2912	285	7071	179
CT	257569	14.7	45056	1951	17.3%	3592	5632	541	1852	219
MR	196608	16.0	65536	1104	1.7%	2210	8192	550	1127	102
Medium	440746	16.0	65536	55839	87.1%	111681	8192	4358	6231	3528
Contrast	440746	16.0	65536	23737	36.4%	47475	8192	1251	23737	909
Gamma	440746	16.0	65536	28076	44.4%	56154	8192	1546	28080	1314
Small	48776	16.0	65536	25174	39.7%	50350	8192	8447	13195	7288

The RLE+LZ77 method appears to be the most efficient, therefore for further evaluating effects of histogram packing on compression ratios of popular algorithms we use the RLE+LZ77 method. The compression ratios obtained for images before histogram packing (Norm.), after packing (Pack.), and the ratio improvements due to histogram packing are reported in the Table 3. The compression ratio is expressed in bits per pixel [bpp]: $8\,e\,/\,n$, where n is the number of pixels in the image, e—the size in bytes of the compressed image (including the size of the histogram encoded using the RLE+LZ77 method in the case of ratio after packing). We performed experiments for the following image compression algorithms: CALIC [8] (implementation by Wu and Memon), JPEG-LS [5] (SPMG/UBC implementation version 2.2), and JPEG2000 [9] (JasPer implementation by Adams version 1.700.0).

TABLE 3. Effects of histogram packing on compression ratios of CALIC, JPEG-LS, and JPEG2000; results obtained for histograms encoded using RLE+LZ77 method (averages for groups).

Images		CALIC			JPEG-LS			JPEG2000		
Name	***U***	**Norm. [bpp]**	**Pack. [bpp]**	**Impro-vement**	**Norm. [bpp]**	**Pack. [bpp]**	**Impro-vement**	**Norm. [bpp]**	**Pack. [bpp]**	**Impro-vement**
CR	59.5%	6.229	5.287	15.1%	6.343	5.398	14.9%	6.394	5.426	15.1%
CT	17.3%	7.759	4.485	42.2%	7.838	4.557	41.9%	8.044	4.630	42.4%
MR	1.7%	9.975	4.811	51.8%	10.009	4.944	50.6%	10.024	4.849	51.6%
Medium	87.1%	11.735	11.760	-0.2%	11.829	11.844	-0.1%	12.058	12.082	-0.2%
Contrast	36.4%	11.330	10.010	11.6%	11.416	9.992	12.5%	11.951	10.558	11.7%
Gamma	44.4%	11.850	10.646	10.2%	11.950	10.676	10.7%	12.183	10.965	10.0%
Small	39.7%	12.547	12.939	-3.1%	12.414	12.813	-3.2%	12.712	13.180	-3.7%

We notice, that effects of packing histograms on the compression ratios of tested algorithms are, for all algorithms, very similar (see also Fig. 1.a). As expected, the histogram packing does not improve compression ratios for Small images. Also the average compression ratio of Medium images, most of which have non-sparse histograms, gets negligibly worse if we employ histogram packing. Except for the above cases, the histogram packing improves average compression ratios for high bit depth sparse histogram images. The improvement varies depending on the image level utilization U, which we use as a measure of the histogram sparseness (see Fig. 1.b). For $U < ¼$ the compression ratio improvement is roughly 50%, i.e., the size of the compressed image gets halved by applying the histogram packing method. For $U \approx ½$ we get the compression ratio improvement of about 10–20%; this level of improvement is not negligible for lossless image compression algorithm—the difference in compression ratio between algorithms obtaining best ratios and algorithms obtaining best speeds usually does not exceed 10% for the images used [7]. For $U > ¾$ the histogram packing improves ratios for some images only, however, it does not deteriorate ratios for the remaining ones.

The greatest improvements are obtained for CT and MR images, yet for these images the use of another histogram encoding method, namely MT, may be a practical alternative. This way, at the cost of loosing of small fraction of the ratio improvement obtained (compare Tables 2 and 3) we get possibility to decompress image within standard algorithms like JPEG-LS (which is included in the DICOM standard [10]) or JPEG2000 (2nd part [11]). Except that ratio improvement for CR images is smaller than for CT and MR, above conclusion applies to medical

CR images also. Using the off-line histogram packing and the MT method of histogram encoding we may significantly improve compression ratios of medical images while maintaining compatibility with current standards.

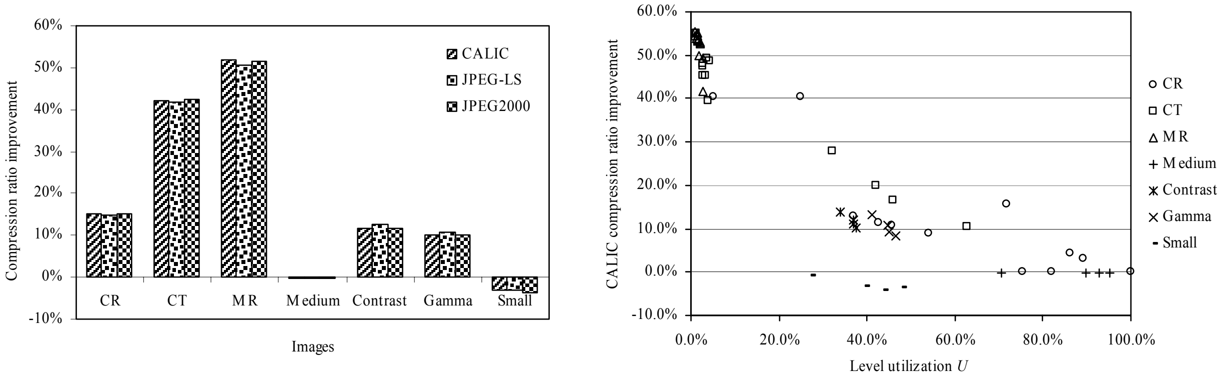

FIGURE 1. a) Average compression ratio improvement due to histogram packing (RLE+LZ77); **b)** CALIC compression ratio improvement of individual images due to histogram packing (RLE+LZ77).

CONCLUSIONS

We introduced and analyzed experimentally a couple of methods of encoding histograms of high bit depth images. One of the methods (RLE+LZ77) obtains the shortest encoded histogram length for nearly all tested images and in practice is sufficiently good for encoding histograms of wide range of images. A simpler method (MT) may be useful for medical images. For these images, its use results in improvements of the compression ratio little worse compared to RLE+LZ77, but decoding of images with packed histograms encoded using the MT method is already supported by JPEG-LS and JPEG2000 (part 2) standards. The effects of packing histograms on the compression ratios of CALIC, JPEG2000, and JPEG-LS are, for all tested algorithms, very similar—histogram packing improves significantly lossless compression ratios for high bit depth sparse histogram images. The ratio improvement due to histogram packing may exceed a factor of two, as in the case of MR medical images.

ACKNOWLEDGMENTS

This research was supported by the grant BK-239/Rau-2/2005 from the Institute of Computer Science, Silesian University of Technology.

REFERENCES

1 Y. Yoo, Y. G. Kwon, A. Ortega*, Embedded image-domain compression of simple images*, Proc. of the 32nd Asilomar Conf. on Signals, Systems, and Computers, 2 (1998) 1256-1260.
2. A. J. Pinho, *On the impact of histogram sparseness on some lossless image compression techniques*, Proc. ICIP-2001, III (2001) 442-445.
3. R. Starosolski, *Compressing images of sparse histograms*, Proc. SPIE Medical Imaging, 5959 (2005) 595912(209-217).
4. WWW Consortium, *W3C Recommendation: PNG (Portable Network Graphics) Specification*, Version 1.0, 1996.
5. ISO/IEC, ITU-T, *Information technology – Lossless and near-lossless compression of continuous-tone still images – Baseline*, ISO/IEC International Standard 14495-1 and ITU-T Recommendation T.87, June 1999.
6. J. Ziv, A. Lempel, *A universal algorithm for sequential data compression*, IEEE Trans. on Inf. Theory, 32(3) (1977) 337-343.
7. R. Starosolski, *Performance evaluation of lossless medical and natural continuous tone image compression algorithms*, Proc. SPIE Medical Imaging, 5959 (2005) 59590L(116-127).
8. X. Wu, N. Memon, *Context-based, Adaptive, Lossless Image Codec*, IEEE Trans. Comm., 45(4) (1997) 437-444.
9. C. Christopoulos, A. Skodras, T. Ebrahimi, *The JPEG2000 Still Image Coding System an Overview*, IEEE Trans. on Consumer Electronics, 46(4) (2000) 1103-1127.
10. National Electrical Manufacturers Association, *Digital Imaging and Communications in Medicine (DICOM) standard*, 2004.
11. ISO/IEC, ITU-T, *Information technology – JPEG 2000 image coding system: Extensions*, ISO/IEC International Standard 15444-2 and ITU-T Recommendation T.801, August 2002.