Performance evaluation of lossless medical and natural continuous tone image compression algorithms

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

ABSTRACT

In this study we evaluate the performance of several lossless grayscale image compression algorithms: algorithms that are standards in medical image transmitting and archiving systems, other algorithms used for compressing medical images in practice and in image compression research, and of a couple of universal algorithms applied to raw and preprocessed image data. In the experiments we use a new, publicly available, test image set, which is described in detail in the paper. The set contains about one hundred images, mainly medical images of various modalities (CR, CT, MR, and US) and natural continuous tone grayscale images of various sizes and various bit depths (up to 16 bits per pixel). We analyze algorithm performance with respect to image modality, depth, and size. Our results generally adhere to results reported in other studies, however, we find that some common opinions on performance of popular algorithms are imprecise, or even false. Most interesting observation concerning the compression speed is that the speed of many algorithms is relatively low, e.g., JPEG2000 obtains speed close to CALIC algorithm, which is considered to be slow. On the other hand there exist algorithms much faster than the JPEG-LS (i.e., SZIP and SFALIC). Considering the compression ratio, the most interesting results were obtained for high bit depth medical CT and MR images, which are of sparse histograms. For better compression ratios of those images, instead of standard image compression algorithms, we should either use universal algorithms or employ the histogram packing technique prior to actual image compression.

Keywords: lossless image compression, medical images, high bit depth images, image compression standards, test images, CALIC, JPEG-LS, JPEG2000, SZIP, SFALIC

1. INTRODUCTION

In this study we evaluate the performance of several lossless grayscale image compression algorithms. In the experiments we use a new, publicly available, test image set, which is described in detail in the paper. The set contains about one hundred images, mainly medical images of various modalities (CR, CT, MR, and US) and natural continuous tone grayscale images of various sizes and various depths (up to 16 bits per pixel). The set contains also non-typical images used to estimate the best case and the worst case performance of compression algorithms (easily compressible images, images with added noise, and ones containing nothing, but the noise).

We analyze the performance of algorithms, which are standards in medical image transmitting and archiving systems, of other algorithms used for compressing medical images in practice and in image compression research, and of a couple of universal algorithms applied to raw and preprocessed image data. We analyze algorithm performance with respect to image modality, depth, and size. As opposed to most of other studies we analyze both the compression ratio and the compression speed. Our results generally adhere to results reported in other studies, however, we find that some common opinions on performance of popular algorithms are imprecise, or even false. The above statement concerns mainly the compression speed, as obtained by standard implementations of image compression algorithms, and the compression ratios obtained for high bit depth images.

The reminder of this paper is organized as follows. In section 2 we describe set of test images, in section 3 we characterize, very briefly, the algorithms we evaluate. Experimental procedure is described in section 4; the obtained results are presented and discussed in section 5. Section 6 contains conclusions of this study.

* Roman.Starosolski@polsl.pl

2. SET OF TEST IMAGES

2.1. General set description

The new set of medical and natural continuous tone grayscale test images was prepared to evaluate the performance of lossless image compression algorithms. The set contains natural continuous tone grayscale images of various bit depths (up to 16 bits), various sizes (up to about 4 millions of pixels) and medical images of various modalities (CR, CT, MR, and US). In the set, image groups were defined, to permit performance analysis based on average results for the whole group, rather than on results for single images. The biggest group, normal, is for evaluating algorithms' performance in a typical case. A collection of smaller groups permits to analyze or compare results with respect to images' bit depths, sizes, or medical image modality. The set contains also non-typical images, which do not belong to the normal group. To analyze the algorithms' performance on noisy data special images with added noise were prepared. To estimate the best-case and the worst-case performance of algorithms, easily compressible and incompressible pseudo-images were also generated. In the following sections we describe image groups, details of individual images are reported in Appendix A. The set contains about one hundred images. It is not as large as, e.g., the set used by Clunie in an extensive study on lossless compression of medical images[1] (over 3600 images), but on the other hand moderate size of the set allowed making it publicly available. The set may be downloaded from http://sun.iinf.polsl.gliwice.pl/~rstaros/mednat/index.htm.

2.2. Natural images

Natural images are continuous tone images acquired from scenes available for human eye (photographic images). The group of natural images was constructed as follows. Four images (Fig. 1) were acquired from a 36mm high quality diapositive film (Fuji Provia/Velvia) using Minolta Dimage 5400 scanner. In order to minimize the noise, the acquisition was first done at device's maximum depth of 16 bits, optical resolution 5400dpi, and using multiple sampling of each pixel. One image ("flower") was softened by setting scanner focus too close. For all images, but one ("branches"), we used the scanner's optical mechanism of reducing visibility of the film grain ("grain dissolver"). Then images' resolution was reduced 3 times. These images formed a group of 16-bit big images, and then were subject to further resolution reduction (3 and 9 times) and to bit depth reduction (to 12 and to 8 bits). The set contains following groups of natural images:

- *natural*—main group of natural images, 36 natural images of various sizes and bit depths,
- *big*—12 natural images of various bit depths and size approximately 4000000 pixels,
- *medium*—12 natural images of various bit depths and size approximately 440000 pixels,
- *small*—12 natural images of various bit depths and size approximately 49000 pixels,
- *16bpp*—12 natural images of various sizes and 16-bit depth,
- *12bpp*—12 natural images of various sizes and 12-bit depth,
- *8bpp*—12 natural images of various sizes and 8-bit depth.

2.3. Medical images

Groups of medical images were composed of CR, CT, MR, and US images of various anatomical regions, acquired from devices of several vendors. Note, that in case of medical CR, CT, and MR images, the nominal bit depth may be misleading. Actual number of intensity levels of these images may be smaller than implied by the bit depth by an order of magnitude or even more—see Appendix A for details. Set contains following groups of medical images:

- *medical*—main group of natural images, 48 medical CR, CT, MR, and US images,
- *cr*—12 medical CR images, nominal depth: 10 to 16 bits, average size approximately 3500000 pixels,
- *ct*—12 medical CT images, nominal depth: 12 to 16 bits, average size approximately 260000 pixels,
- *mr*—12 medical MR images, nominal depth of 16 bits, average size approximately 200000 pixels,
- *us*—12 medical US images, 8-bit depth, average size approximately 300000 pixels.

2.4. The *normal* group

The *normal* group is a main group of the set; it contains all 84 images from groups *natural* and *medical*. The *normal* group is for evaluating algorithms' performance in a typical case, i.e., average results of compressing images from the *normal* group may serve as a measure of algorithm's average performance for continuous tone grayscale images.


“branches”


“flower”


“kid”


“town”

Fig. 1. Sample *natural* images.

2.5. Non-typical images

Following groups of non-typical images are contained in the set:

- *noise*—9 images with added noise, created using “branches” image of various bit depths (8, 12, and 16 bits) and medium size (approximately 440000 pixels). Noise was added using: $v_1 = v_0(1 - a) + r\,a$, where v_0 denotes original pixel intensity, v_1—intensity after adding noise, r—random value of uniform distribution, and a is the amount of noise. We prepared images using $a = 0.1$, 0.2, and 0.5,
- *empty*—3 pseudo-images, intensity of all pixels equals 0, nominal depth of 8, 12, and 16 bits, size approximately 440000 pixels,
- *random*—3 pseudo-images, random intensities of pixels (uniform distribution), bit depth of 8, 12, and 16 bits, size approximately 440000 pixels.

The *random* pseudo-images may be used to evaluate the worst-case performance of an image compression algorithm, however modern image compression algorithms are based on sophisticated assumptions as to characteristics of data they process. For a specific image compression algorithm we could prepare data even harder to compress, i.e., pseudo-image of characteristics opposite to what is expected in the compression algorithm.

3. ALGORITHMS AND IMPLEMENTATIONS

In this section we characterize briefly the algorithms analyzed in this study. In the experiments, we have used about ten algorithms. Due to the number of algorithms tested, the more detailed description of them exceeds the scope of this paper. In section 5 we report results of the following image compression algorithms and implementations (using the default options, unless indicated otherwise):

- Lossless JPEG—former JPEG committee standard for lossless image compression[2]. The standard describes predictive image compression algorithm with Huffman[3] or arithmetic[4] entropy coder. We used the PVRG-JPEG implementation, version 1.2.1[5]. The implementation uses Huffman codes. The results are reported for the predictor function SV 4, which results in the best average compression ratio of *normal* images.
- JPEG-LS—standard of the JPEG committee for lossless and near-lossless compression of still images[6]. The standard describes low-complexity predictive image compression algorithm with entropy coding using modified Golomb-Rice[7, 8] family. The algorithm is based on the LOCO-I algorithm[9, 10]. We used the SPMG/UBC implementation[11].

- JPEG2000—a recent JPEG committee standard describing algorithm based on wavelet transform image decomposition and arithmetic coding[12]. Apart from lossy and lossless compressing and decompressing of whole images it delivers many interesting features (progressive transmission, region of interest coding, etc.)[13]. We used JasPer implementation by Adams[14].
- PNG—standard of the WWW Consortium for lossless image compression[15]. PNG is a predictive image compression algorithm, using the LZ77[16] algorithm and the Huffman codes. We used pnmtopng implementation, version 2.37.6[17] (NetPBM 10.25, LibPNG 1.28, ZLIB 1.22). The results are reported for the Paeth predictor function (filter), which results in the best average compression ratio of *normal* images. We used the fastest (-compression 1) speed option, which for *normal* images results in doubling the compression speed at the cost of worsening the compression ratio by 4.5% compared to default compression speed setting.
- SZIP—standard of the Consultative Committee for Space Data Systems used by space agencies for compressing scientific data transmitted from satellites and other space instruments[18]. SZIP is a very fast predictive compression algorithm based on the extended-Rice algorithm; it uses Golomb-Rice codes for entropy coding. We used UNM implementation[19]. It's default block size is 16 symbols. Since biggest images (*big* and *cr*) require a greater block size, we used block size of 20 symbols for all the images.
- CALIC-A—a relatively complex predictive image compression algorithm using arithmetic entropy coder, which because of the very good compression ratios is commonly used as a reference for other image compression algorithms[20, 21]. We used implementation by Wu and Memon[22].
- CALIC-H—variant of CALIC algorithm using Huffman codes[22].
- SFALIC—very fast predictive lossless image compression algorithm by the author of this study[23] using modified Golomb-Rice family for entropy coding. We used own implementation[24].
- FELICS—very simple and fast lossless image compression algorithm by Howard and Vitter[25] using Golomb [Gol'1966] or Golomb-Rice codes for entropy coding. We used implementation from the *mg* 1.2.1 system by Bell, Moffat, Witten, and others[26].

Apart from the image compression algorithms, we analyze performance of a couple of universal algorithms. In case of universal algorithms we report both the results obtained by applying the algorithm to raw image data observed in the raster scan order, and the results of compressing images after applying a simple prediction to them. We apply the same prediction as used by default by the SFALIC algorithm—we predict, that the pixel intensity equals $\frac{3}{4}A + \frac{3}{4}B - \frac{1}{2}C$, where A, B, and C are the intenities of pixel's neighbors, respetively: left, upper, and upper-left, then we actually compress the prediction error. We tested the following:

- GZIP—simple and fast universal data compression utility by Gailly. GZIP is based on the LZ77 algorithm. We used implementation version 1.2.4[27].
- BZIP2—universal data compression utility by Seward. BZIP2 is based on the Burrows-Wheeler Block Sorting compression algorithm[28]. We used implementation version 1.0.2[29].

4. PROCEDURE

An HP Proliant ML350G3 computer equipped with two Intel Xeon 3.06 GHz (512 kB cache memory) processors and Windows 2003 operating system was used to measure the performance of algorithm implementations. Single-threaded application of algorithms used for comparisons, were compiled using Intel C++ 8.1 compiler. To minimize effects of the system load and the input-output subsystem performance on obtained results the executable was run several times. The time of the first run was ignored and the collective time of other runs (executed for at least one second, and at least 5 times) was measured and then averaged. The time measured was a sum of time spent by the processor in an application code and in kernel functions called by the application, as reported by the operating system after application execution. Since we measure the execution time externally, we actually include the time of initializing the program by the operating system into our calculations; this time may be significant for smallest images. In case of the CALIC algorithm implementation, that is available as a binary executable for UltraSparc processors, the speed is estimated based on the relative speed of this implementation, as compared to the SFALIC speed, on another computer system (Sun Fire V440 running Solaris 9, equipped with 1.06GHz UltraSparc IIIi processors; both implementations were single-threaded).

The compression speed is reported in megabytes per second [MB/s], where 1MB = 2^{20} bytes. Since we used PGM P5 image representation, the pixel size is 2 bytes for image depth over 8 bits, 1 byte in the opposite case. The compression ratio is in bits per pixel [bpp]: $8e/n$, where e is the size in bytes of the compressed image including the header, n—number of pixels in the image.

5. RESULTS

For *natural* images, the compression ratios (Table 1) obtained by the tested algorithms generally adhere to results reported by other studies and to common opinions on ratios of popular algorithms. CALIC-A obtains the best average ratio, it's Huffman coder version along with the JPEG-LS obtain ratios worse by less than 1%. JPEG2000 and SFALIC obtain ratios worse by about 4%. Ratios of Lossless JPEG and SZIP are by about 10% worse. Relative ratio of those algorithms compared to CALIC-A depends significantly on image bit depth. For high bit depth images the differences in ratios are significantly smaller, than for low bit depths. Interestingly, for some *16bpp* images (and also for groups: *small* and *mr*) the Huffman coder version of CALIC obtains ratios better, than the arithmetic coder version—probably there is still a possibility to improve this algorithm.

Algorithms PNG and FELICS obtain ratios worse to general purpose BZIP2 algorithm and to simple general-purpose GZIP, as applied to image after prediction. PNG performs poorly regardless of images sizes and depths. For 8-bit images FELICS is better than Lossless JPEG, for high bit depths, the variant of FELICS we examined causes huge data expansion. Universal algorithms perform much better, when applied to images after prediction. The compression ratio of BZIP2 is close to ratios of SZIP and Lossless JPEG, ratio of GZIP is worse, but still better than the ratio of PNG.

Table 1. The compression ratio, natural and medical images [bpp].

image group	**Lossless JPEG**	**JPEG-LS**	**JPEG2000**	**PNG**	**SZIP**	**CALIC-A**	**CALIC-H**
natural	8.367	7.687	7.916	10.045	8.432	7.617	7.662
big	7.668	7.083	7.185	9.451	7.773	6.962	7.059
medium	8.446	7.710	7.955	10.079	8.403	7.623	7.699
small	8.986	8.269	8.608	10.605	9.121	8.267	8.227
16bpp	12.327	11.776	11.998	13.836	12.459	11.748	11.622
12bpp	8.321	7.571	7.823	11.542	8.407	7.491	7.565
8bpp	4.451	3.715	3.927	4.756	4.431	3.613	3.797
medical	7.427	6.734	6.891	8.073	7.396	6.651	6.761
cr	7.023	6.343	6.394	8.944	6.883	6.229	6.324
ct	8.509	7.838	8.044	9.381	8.806	7.759	7.840
mr	10.451	10.009	10.024	10.350	10.599	9.975	9.895
us	3.724	2.748	3.100	3.616	3.298	2.641	2.985
normal	7.830	7.143	7.330	8.918	7.840	7.065	7.147

image group	**SFALIC**	**FELICS**	**GZIP *-pred***	**GZIP**	**BZIP2 *-pred***	**BZIP2**
natural	7.953	62.359	9.325	10.866	8.436	9.165
big	7.274	9.396	8.688	10.598	7.603	8.360
medium	8.009	34.954	9.387	10.800	8.487	9.113
small	8.576	142.727	9.899	11.198	9.219	10.023
16bpp	11.867	172.659	13.537	15.185	12.687	14.059
12bpp	7.869	10.277	10.030	11.843	8.331	8.877
8bpp	4.123	4.141	4.408	5.569	4.290	4.560
medical	7.165	11.168	7.757	7.770	6.378	5.181
cr	6.662	6.971	8.038	9.812	6.696	6.479
ct	8.266	14.461	9.155	8.678	7.389	5.577
mr	10.235	19.847	10.532	9.098	8.375	5.929
us	3.497	3.394	3.301	3.492	3.051	2.739
normal	7.503	33.107	8.429	9.097	7.260	6.889

As to *medical* images, they differ from *natural* ones in two ways. Some of them (*us* and, to some extent, *mr* and *ct*) are compound images containing large uniform intensity areas, i.e., background for the actual medical image. Some of the tested algorithms are able to encode such areas with ratio smaller than 1 bpp, others (Lossless JPEG, SFALIC, and FELICS) lack such mechanism—in case of *us* images, those algorithms obtain ratios worse than CALIC-A by about 30 to 40%.

The second characteristic feature is common for *mr* and *ct* images. These images are of high nominal bit depth, which is from 12 to 16 bits per pixel. The actual number of pixels' intensity levels found in those images is smaller, than implied by the nominal bit depth, sometimes by an order of magnitude or even more (see Table A in Appendix A). Furthermore, intensity levels are distributed throughout the entire nominal intensity range, i.e., the images have sparse histograms of intensity levels. None of the image compression algorithms analyzed in this study was designed for images of sparse histograms.

For *natural* images the universal compression algorithm BZIP2 obtains ratios worse than CALIC by about 10%. For medical *mr* images, that all are of 16-bit nominal depth, and that none of them actually contains pixels of more than 2000 levels, ratio of BZIP2 is better than CALIC's by over 40%. In case of *mr* images, ratios of the very simple universal algorithm GZIP, are also better then ratios of the best image compression algorithms. Furthermore, the prediction, that improves the BZIP2 ratio for *normal* images, deteriorates it significantly for *mr* and *ct* images (similar behavior is observed for the GZIP algorithm). For *cr* and *us* images the best ratios were obtained by CALIC, however, the compression ratio deterioration of *mr* and *ct* images is so high, that the average compression ratio of the whole *medical* group, and of the whole *normal* group, is best in case of the BZIP2.

The impact of histogram sparseness on image compression ratios has been recently discovered[30, 31]. Most research, however, was done for low bit depth images—from results presented in Table 1 we conclude, that also in case of high bit depth medical images, histogram sparseness deteriorates the compression ratio. Above observations triggered further research on methods of improving the compression ratios of high bit depth sparse histogram images[32]. It was found, that the so-called histogram packing technique vastly improves compression ratios—the CALIC average compression ratios got improved to 4.485 bpp for *ct* and to 4.811 bpp for *mr* images. These results may be surprising, however, they clearly indicate that, in case of certain modalities, for better compression ratios instead of image compression algorithms we should either use universal algorithms or employ the histogram packing technique prior to actual image compression.

The compression speed of *normal* images is presented in Table 2. The average speed and ratio for the *normal* group is presented on the Figure 2. The speed results are implementation dependent. We used standard implementations of all the algorithms, and (unless indicated otherwise) standard options. The speed of some of the algorithms could be improved. The Lossless JPEG implementation we used is not optimized for speed. According to Santa-Cruz and Ebrahimi[33] the speed of another Lossless JPEG implementation (Huang, Smith, Cornell University, version 1.0) is for 8-bit images about 2 times lower, than the speed of JPEG-LS—we do not report results of that implementation, since it is not lossless for high bit depth images. The speed of GZIP may be increased about two times by using the fastest (--fast) compression option—this way we worsen average compression ratio of *normal* images by 2.2%. The speed of CALIC was estimated, to verify this estimation we measured the speed of another implementation (by Yuan, designed for 8-bit images only and probably not optimized for speed). Compared to the speed of CALIC-A reported in Table 2, the compression speed of Yuan's CALIC for *8bpp* and *us* images was lower by about 25%. Measuring speed of GZIP and BZIP2, as applied to images after prediction, we ignored the time required to perform the prediction—this time may not be negligible, since in case of SFALIC algorithm, about 20% of compression time is spent on performing the prediction.

Analyzing the average compression speed of *normal* images, we find that some common opinions on speed of popular algorithms are imprecise or even false. Some algorithms obtain relatively low compression speed compared to the CALIC algorithm, which is considered to be slow. JPEG2000 obtains speed little lower, than the speed of CALIC-A, PNG's speed is little lower then the speed of CALIC-H. On the other hand there are significant speed differences among algorithms commonly referred as fast: FELICS, SZIP, and JPEG-LS. Only for 8-bit images the speed of FELICS is close to the speed of JPEG-LS. For all the groups of *normal* images algorithms SZIP and SFALIC obtain speed significantly higher, than the JPEG-LS. They are, on average, 2.5 times faster then the JPEG-LS. The speed of SFALIC and SZIP is almost identical, except for 8-bit images (both *8bpp* and *us* groups), where SFALIC is faster by 10-20%.

Table 2. The compression speed, natural and medical images [MB/s].

image group	Lossless JPEG	JPEG-LS	JPEG2000	PNG	SZIP	CALIC-A	CALIC-H
natural	4.8	15.0	2.7	6.4	40.3	2.6	7.1
big	5.4	18.0	3.0	7.0	56.0	3.3	9.7
medium	5.2	16.4	2.8	6.7	44.8	2.7	8.0
small	3.8	10.6	2.3	5.4	20.0	1.6	3.7
16bpp	5.4	16.1	2.4	6.8	40.0	1.9	6.9
12bpp	5.8	17.7	3.2	7.1	48.4	3.0	7.5
8bpp	3.2	11.2	2.6	5.2	32.4	3.1	7.1
medical	5.5	20.6	2.8	7.9	50.6	3.5	9.4
cr	6.7	25.0	4.1	8.9	73.5	4.7	11.4
ct	5.7	21.4	2.3	8.5	50.1	3.3	9.1
mr	5.7	20.5	2.2	8.2	41.6	2.6	8.1
us	3.7	15.7	2.8	5.9	37.1	3.7	9.3
normal	5.2	18.2	2.8	7.2	46.1	3.1	8.4
image group		**SFALIC**	**FELICS**	**GZIP -pred**	**GZIP**	**BZIP2 -pred**	**BZIP2**
natural		43.3	8.7	6.0	8.1	2.6	2.7
big		61.8	11.4	6.0	9.0	2.2	2.5
medium		48.1	9.1	6.4	8.5	2.4	2.5
small		20.1	5.7	5.7	6.8	3.2	2.9
16bpp		41.3	5.2	7.8	9.2	2.1	2.0
12bpp		47.9	10.7	5.0	8.1	2.7	2.7
8bpp		40.7	10.3	5.2	7.0	2.9	3.3
medical		50.1	13.1	7.7	9.5	3.6	3.5
cr		70.5	16.8	6.1	10.2	2.5	2.6
ct		49.1	12.0	7.3	9.0	3.5	2.6
mr		39.9	10.4	9.6	9.9	3.3	3.3
us		40.8	13.1	8.0	9.0	5.1	5.7
normal		47.2	11.2	7.0	8.9	3.2	3.2

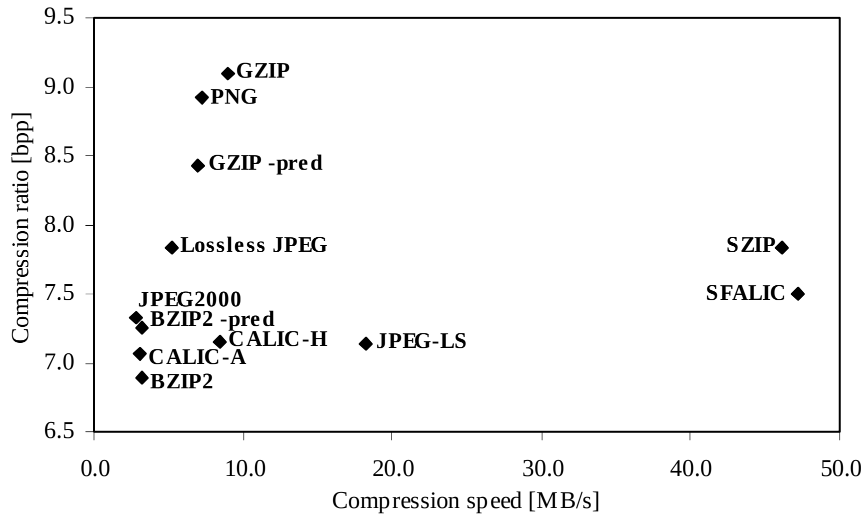

Fig. 2. Average compression speed and ratio for the *normal* group.

Table 3. The compression speed, non-typical images [MB/s].

image group	Lossless JPEG	JPEG-LS	JPEG2000	PNG	SZIP	CALIC-A	CALIC-H
empty	6.5	157.0	11.3	21.8	118.2	13.1	39.8
noise	5.0	15.2	2.3	6.4	41.4	2.0	7.4
random	5.0	15.2	2.0	6.5	41.5	1.3	6.5
image group		**SFALIC**	**FELICS**	**GZIP *-pred***	**GZIP**	**BZIP2 *-pred***	**BZIP2**
empty		79.3	69.8	68.1	74.3	40.8	40.6
noise		45.6	6.7	9.6	9.5	1.8	1.9
random		37.3	4.6	13.0	12.7	1.5	1.5

Table 4. The compression ratio, non-typical images [bpp].

image group	Lossless JPEG	JPEG-LS	JPEG2000	PNG	SZIP	CALIC-A	CALIC-H
empty	1.001	0.002	0.004	0.065	0.027	0.001	0.045
noise	11.252	10.693	10.841	12.348	11.101	10.478	10.690
random	12.976	12.516	12.891	13.368	12.370	12.375	13.008
image group		**SFALIC**	**FELICS**	**GZIP *-pred***	**GZIP**	**BZIP2 *-pred***	**BZIP2**
empty		1.000	1.000	0.014	0.014	0.001	0.001
noise		10.842	73.055	11.682	11.886	11.135	11.177
random		12.009	516.440	12.574	12.573	12.102	12.111

In case of fast algorithms (JPEG-LS, SZIP, and SFALIC) significantly lower speed, than the average, was obtained for *small* images. For those images time of initializing the compression implementation executable by the operating system starts to be a significant factor of the overall speed of the compression algorithm. To some extent similar behavior may be observed for all of the examined algorithms, but the BZIP2. Since for depths over 8 bits the image pixel is stored using 2 bytes, the compression speed of *12bpp* images is usually greater than the speed of *8bpp* and *16bpp* images.

The compression results of non-typical images are presented in Tables 3 and 4. The *empty* pseudo-images are the most easily compressible data for the image compression algorithm. As one could expect, for *empty* images the ratio of algorithms that employ a method of efficient encoding of uniform image regions is close to 0 bpp. For all the algorithms the compression speed of *empty* group is higher than of any other group, the greatest speedup is observed for JPEG-LS.

For non-typical noisy images the compression speed of all algorithms, but the GZIP, is little lower than the average *medium* group speed that contains images of similar size. In case of all algorithms, but PNG and FELICS, compression of those images, in case of some algorithms even of individual images with 50% noise added, still results in compression ratios smaller than the image bit depth, however not by much. The *random* pseudo-images are incompressible and may be used for estimating the worst-case algorithm compression ratio. The best method of processing incompressible data is to copy them binary, i.e., to encode pixel intensities using N-bit natural binary code, where N denotes image bit depth—for the *random* group we would get the compression ratio of 12 bpp. The SFALIC algorithm actually acts this way; all the remaining algorithms cause noticeable data expansion, which, except for FELICS and PNG, ranges from 0.1 to 1 bit per pixel.

6. CONCLUSIONS

In this study, we analyzed compression ratio and compression speed of algorithms, we did not consider other properties of algorithms, like: ability of progressive coding, region of interest coding, existence of standard describing he algorithm, availability of implementation, or popularity of an algorithm. We have tested image compression algorithms (Lossless JPEG, JPEG-LS, JPEG2000, PNG, SZIP, CALIC, SFALIC, and FELICS) as well as the universal algorithms (GZIP, BZIP2). Based on two criteria only, we find that there is no single algorithm best suitable for all the image classes. Depending on the preference of the user (speed, ratio, or both) different algorithms are good choices for the different image groups.

The experiments were performed using a new set of medical and natural continuous-tone images. The best average compression ratios for natural continuous-tone images and for CR and US medical images are obtained by the CALIC algorithm using arithmetic entropy coder. In case of remaining medical images, i.e, MR and CT, the CALIC is best among image compression algorithms, but surprisingly much better ratios were obtained by a universal algorithm—BZIP2. For those images, the difference in ratios between CALIC and BZIP2 is so high, that not only the average compression ratio of all the medical images, but the average ratio of whole group of medical and natural images as well, is best in case of the BZIP2. For the MR images the ratio of BZIP2 is better than CALIC's by over 40%. CT and MR images are of sparse histograms, the tested image compression algorithms were not designed to process such data. For better compression ratios of CT and MR images, instead of image compression algorithms, we should either use universal algorithms or employ the so-called histogram packing technique prior to actual image compression.

On a computer equipped with a 3.06 MHz Intel Xeon processor, both CALIC and BZIP2, obtain compression speed of about 3 MB/s. If we need a faster compression algorithm, then we should use JPEG-LS, SFALIC, or SZIP. JPEG-LS obtains average compression speed of over 18 MB/s, it's average compression ratio is worse than CALIC's by about 1%. SFALIC and SZIP obtain average compression speed of over 45 MB/s, i.e., compression speed of those algorithms is about 2.5 times higher, than JPEG-LS's, and about 15 times higher, than the speed of CALIC or BZIP2. Compared to CALIC, the average compression ratio of SFALIC and SZIP is worse by about 6% and 11% respectively.

We notice, that some algorithms obtain relatively low compression speed. JPEG2000 obtains speed little lower, than the speed of arithmetic coder version of CALIC, which is considered to be slow, PNG's speed is little lower then the speed of Huffman coder version of CALIC. We also find that some algorithms capable of processing high bit depth images, like PNG and FELICS, actually are not suitable for that purpose, because of poor compression ratios. For high bit depth natural continuous-tone images we notice two more interesting facts. Firstly, for those images the best ratio was obtained by the Huffman coder version of CALIC, not the arithmetic coder version. Also, the differences in compression ratios of various algorithms are for 16-bit images much smaller, than for 8-bit ones—the difference in compression ratio between the fastest algorithm (SFALIC) and the one obtaining best ratios (CALIC Huffman) is about 2% only.

ACKNOWLEDGMENTS

The research was fully supported by the Grant Nr 4 T11C 032 24 of the Polish National Research Committee (KBN). Above Grant was carried out at the Institute of Computer Science, Silesian University of Technology, in years 2003 and 2004. The author would like to thank people and institutions, which in helped in obtaining test images and kindly agreed to include the images in the set. Natural images are scanned photographs by Dr Jacek Szedel (Silesian University of Technology). Some of the medical images were supplied by Prof. Ewa Piętka (Silesian University of Technology), some others are Philips DICOM Reference Medical Images of Philips Medical Systems, all the remaining ones are the publicly available DICOM images, donated by various medical equipment vendors to the RSNA conference.

BIBLIOGRAPHY

1. D. A. Clunie, "Lossless compression of grayscale medical images - effectiveness of traditional and state of the art approaches", *Proc. SPIE, Medical Imaging 2000*, **3980**:74-84, 2000.
2. G. Langdon, A. Gulati, E. Seiler, "On the JPEG model for lossless image compression", *Proceedings DCC '92, Data Compression Conference*, 172-80, 1992.
3. D. A. Huffman, "A method for the construction of minimum-redundancy codes", *Proceedings of the Institute of Radio Engineers*, **40**(9):1098-101, 1952.
4. A. Moffat, R. M. Neal, I. H. Witten, "Arithmetic Coding Revisited", *ACM Transactions on Information Systems*, **16**(3):256-94, 1998.
5. Portable Video Research Group (PVRG), Stanford University, "PVRG-JPEG codec", version 1.2.1, 1995, http://www.dclunie.com/jpegge/JPEGv1.2.1.tar.gz.
6. ISO/IEC, "Lossless and near-lossless compression of continuous-tone still images (JPEG-LS)", ISO/IEC FCD 14495-1 – public draft, 1997.

7. S. W. Golomb, “Run-Length Encodings” *IEEE Trans. on Information Theory*, **IT-12**:399-401, 1966.
8. R. F. Rice, “Some practical universal noiseless coding techniques – part III”, Jet Propulsion Laboratory tech. report JPL-79-22, 1979.
9. M. J. Weinberger, G. Seroussi, G. Sapiro, “LOCO-I: A low complexity, context based, lossless image compression algorithm”, *Proceedings DCC'96*, 140-9, 1996.
10. M. J. Weinberger, G. Seroussi, G. Sapiro, “The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS”, *IEEE Trans. Image Processing*, **9**(8):1309-24, 2000.
11. Signal Processing and Multimedia Group, Univ. of British Columbia, “SPMG/JPEG-LS implementation”, version 2.2, 1999, ftp://ftp.netbsd.org/pub/NetBSD/packages/distfiles/jpeg_ls_v2.2.tar.gz.
12. ISO/IEC, “JPEG2000 Part I Final Committee Draft Version 1.0: JPEG2000 Image Coding System”, ISO/IEC FCD 15444-1, 2000.
13. C. Christopoulos, A. Skodras, T. Ebrahimi, “The JPEG2000 Still Image Coding System an Overview”, *IEEE Transactions on Consumer Electronics*, **46**(4):1103-27, 2000.
14. M. D. Adams, “JasPer Software Reference Manual and Implementation”, version 1.700.0, 2003, http://www.ece.uvic.ca/~mdadams/jasper/.
15. WWW Consortium, “W3C Recommendation: PNG (Portable Network Graphics) Specification”, Version 1.0, 1996, http://www.w3.org/TR/REC-png.html.
16. J. Ziv, A. Lempel, “A universal algorithm for sequential data compression”, *IEEE Transactions on Information Theory*, **32**(3):337-43, 1977.
17. A. Lehmann, W. van Schaik, G. Roelofs, “pnmtopng utility”, version 2.37.6 (part of the the NetPBM 10.25 toolkit, compiled with LibPNG 1.2.8 and ZLIB 1.2.2), 2004, http://netpbm.sourceforge.net,.
18. Consultative Committee for Space Data Systems, “Lossless Data Compression. CCSDS Recommendation for Space System Data Standards”, CCSDS 121.0-B-1 Blue Book, 1997.
19. University of New Mexico, Microelectronics Research Center “SZIP science data lossless compression program”, combined version: 1.5, 2002, ftp://ftp.ncsa.uiuc.edu/HDF/lib-external/szip/.
20. X. Wu, N. Memon, “Context-based, Adaptive, Lossless Image Codec”, *IEEE Transactions on Communications*, **45**(4):437-44, 1997.
21. X. Wu, “Efficient Lossless Compression of Continuous-tone Images via Context Selection and Quantization”, *IEEE Trans. on Image Processing*, **IP-6**: 656-64, 1997.
22. X. Wu, N. Memon, “Implementation of Context-based, Adaptive, Lossless Image Coder (CALIC)”, 1995, ftp://ftp.csd.uwo.ca/pub/from_wu/.
23. R. Starosolski, “Simple Fast and Adaptive Lossless Image Compression Algorithm”, 2005 (in review).
24. R. Starosolski, “SFALIC implementation”, version 03, 2005, http://sun.iinf.polsl.gliwice.pl/~rstaros/sfalic/.
25. P. G. Howard, J. S. Vitter, “Fast and efficient lossless image compression”, *Proceedings DCC '93 Data Compression Conference*, 351-60, 1993.
26. T. C. Bell, A. Moffat, I. Witten, J. Zobel, S. Inglis, C. Nevill-Manning, N. Sharman, T. Shimmin, Dept. of Computer Science & Software Engineering, University of Melbourne “The MG Information Retrieval System and documentation”, version 1.2.1, 1995, http://www.cs.mu.oz.au/mg/mg-1.2.1.tar.gz.
27. J.-L. Gailly, “GNU zip (gzip) compression utility”, version 1.2.4, 1993, http://www.gzip.org/.
28. M. Burrows, D. J. Wheeler, “A block-sorting lossless data compression algorithm”, Digital Equipment Corporation, SRC Research Report 124, 1994.
29. J. R. Seward, “BZIP2 a block-sorting file compressor”, version 1.0.2, 2002, http://www.bzip.org/.
30. Y. Yoo, Y. G. Kwon, A. Ortega, “Embedded image-domain compression of simple images”, *Proc. of the 32nd Asilomar Conf. on Signals, Sys., and Computers*, **2**:1256-60, 1998.
31. A. J. Pinho, “A comparison of methods for improving the lossless compression of images with sparse histograms”, *Proc. IEEE Int. Conf. on Image Processing, ICIP-2002*, **2**:673-6, 2002.
32. R. Starosolski, “Compressing Images of Sparse Histograms”, 2005 (in press).
33. D. Santa-Cruz, T. Ebrahimi, “A Study of JPEG2000 Still Image Coding Versus Other Standards”, *Proc. X European Signal Processing Conf. EUSPICO*, **2**: 673-76, 2000.

APPENDIX A. THE TEST IMAGE SET DETAILS

The Table A contains basic characteristics of all the images contained in the set. In this table we also report the actual number of pixel intensity levels (levels). The set is publicly available and may be downloaded from http://sun.iinf.polsl.gliwice.pl/~rstaros/mednat/index.htm.

Table A. Image details.

image	width	height	pixels	bpp	levels	origin	groups
im_branches_03_16	2458	1610	3957380	16	62207	JS	*big, 16bpp, natural, normal*
im_flower_03_16	2458	1610	3957380	16	65191	JS	*big, 16bpp, natural, normal*
im_kid_03_16	2474	1621	4010354	16	64393	JS	*big, 16bpp, natural, normal*
im_town_03_16	2464	1610	3967040	16	64528	JS	*big, 16bpp, natural, normal*
im_branches_09_16	819	536	438984	16	58919	JS	*medium, 16bpp, natural, normal*
im_flower_09_16	819	536	438984	16	46033	JS	*medium, 16bpp, natural, normal*
im_kid_09_16	824	540	444960	16	60421	JS	*medium, 16bpp, natural, normal*
im_town_09_16	821	536	440056	16	57984	JS	*medium, 16bpp, natural, normal*
im_branches_27_16	273	178	48594	16	29543	JS	*small, 16bpp, natural, normal*
im_flower_27_16	273	178	48594	16	17840	JS	*small, 16bpp, natural, normal*
im_kid_27_16	274	180	49320	16	25348	JS	*small, 16bpp, natural, normal*
im_town_27_16	273	178	48594	16	27965	JS	*small, 16bpp, natural, normal*
im_branches_03_12	2458	1610	3957380	12	3906	JS	*big, 12bpp, natural, normal*
im_flower_03_12	2458	1610	3957380	12	4081	JS	*big, 12bpp, natural, normal*
im_kid_03_12	2474	1621	4010354	12	4076	JS	*big, 12bpp, natural, normal*
im_town_03_12	2464	1610	3967040	12	4066	JS	*big, 12bpp, natural, normal*
im_branches_09_12	819	536	438984	12	3870	JS	*medium, 12bpp, natural, normal*
im_flower_09_12	819	536	438984	12	4077	JS	*medium, 12bpp, natural, normal*
im_kid_09_12	824	540	444960	12	4037	JS	*medium, 12bpp, natural, normal*
im_town_09_12	821	536	440056	12	4020	JS	*medium, 12bpp, natural, normal*
im_branches_27_12	273	178	48594	12	3728	JS	*small, 12bpp, natural, normal*
im_flower_27_12	273	178	48594	12	3438	JS	*small, 12bpp, natural, normal*
im_kid_27_12	274	180	49320	12	3918	JS	*small, 12bpp, natural, normal*
im_town_27_12	273	178	48594	12	3818	JS	*small, 12bpp, natural, normal*
im_branches_03_08	2458	1610	3957380	8	246	JS	*big, 8bpp, natural, normal*
im_flower_03_08	2458	1610	3957380	8	256	JS	*big, 8bpp, natural, normal*
im_kid_03_08	2474	1621	4010354	8	255	JS	*big, 8bpp, natural, normal*
im_town_03_08	2464	1610	3967040	8	255	JS	*big, 8bpp, natural, normal*
im_branches_09_08	819	536	438984	8	242	JS	*medium, 8bpp, natural, normal*
im_flower_09_08	819	536	438984	8	256	JS	*medium, 8bpp, natural, normal*
im_kid_09_08	824	540	444960	8	255	JS	*medium, 8bpp, natural, normal*
im_town_09_08	821	536	440056	8	253	JS	*medium, 8bpp, natural, normal*
im_branches_27_08	273	178	48594	8	241	JS	*small, 8bpp, natural, normal*
im_flower_27_08	273	178	48594	8	255	JS	*small, 8bpp, natural, normal*
im_kid_27_08	274	180	49320	8	249	JS	*small, 8bpp, natural, normal*
im_town_27_08	273	178	48594	8	250	JS	*small, 8bpp, natural, normal*
cr_17218	2392	1792	4286464	12	2068	RS	*cr, medical, normal*
cr_17220	2500	2048	5120000	12	3186	RS	*cr, medical, normal*
cr_17222	1792	2392	4286464	12	2939	RS	*cr, medical, normal*
cr_4503	1670	2010	3356700	10	256	RS	*cr, medical, normal*
cr_4507	1760	1760	3097600	10	1024	RS	*cr, medical, normal*
cr_4509	1760	2140	3766400	10	882	RS	*cr, medical, normal*
cr_pacem_1	1716	1910	3277560	16	24180	PH	*cr, medical, normal*
cr_pacem_2	1531	1965	3008415	16	28627	PH	*cr, medical, normal*
cr_rtg_jb	612	746	456552	16	3280	EP	*cr, medical, normal*
cr_siem_01_02	1744	2128	3711232	10	913	RS	*cr, medical, normal*
cr_siem_14_02	1760	2368	4167680	10	638	RS	*cr, medical, normal*
cr_slim_1	1866	2031	3789846	16	26539	PH	*cr, medical, normal*

image	width	height	pixels	bpp	levels	origin	groups
ct_135960_001	512	512	262144	16	2442	EP	*ct, medical, normal*
ct_135960_005	512	512	262144	16	2806	EP	*ct, medical, normal*
ct_17	512	512	262144	12	1883	RS	*ct, medical, normal*
ct_27154	512	512	262144	12	1300	RS	*ct, medical, normal*
ct_29513	340	340	115600	12	2570	RS	*ct, medical, normal*
ct_29920	512	512	262144	12	1723	RS	*ct, medical, normal*
ct_3030	512	691	353792	16	778	RS	*ct, medical, normal*
ct_3071	512	512	262144	16	1696	RS	*ct, medical, normal*
ct_4006	512	512	262144	16	2100	RS	*ct, medical, normal*
ct_4087	512	512	262144	16	1731	RS	*ct, medical, normal*
ct_4165	512	512	262144	16	1735	RS	*ct, medical, normal*
ct_tk_kl_piers0021	512	512	262144	16	2644	EP	*ct, medical, normal*
mr_2321	512	512	262144	16	894	RS	*mr, medical, normal*
mr_2331	512	512	262144	16	893	RS	*mr, medical, normal*
mr_2337	512	512	262144	16	1047	RS	*mr, medical, normal*
mr_2371	512	512	262144	16	1415	RS	*mr, medical, normal*
mr_2412	512	512	262144	16	1300	RS	*mr, medical, normal*
mr_2807	256	256	65536	16	1858	RS	*mr, medical, normal*
mr_2882	512	512	262144	16	501	RS	*mr, medical, normal*
mr_2896	512	512	262144	16	604	RS	*mr, medical, normal*
mr_6624	256	256	65536	16	795	RS	*mr, medical, normal*
mr_6706	256	256	65536	16	1088	RS	*mr, medical, normal*
mr_6774	512	512	262144	16	1799	RS	*mr, medical, normal*
mr_6837	256	256	65536	16	1055	RS	*mr, medical, normal*
us_19773	640	480	307200	8	256	RS	*us, medical, normal*
us_27704	640	480	307200	8	249	RS	*us, medical, normal*
us_27743	640	480	307200	8	246	RS	*us, medical, normal*
us_28279	640	480	307200	8	250	RS	*us, medical, normal*
us_28282	640	480	307200	8	247	RS	*us, medical, normal*
us_28289	640	480	307200	8	254	RS	*us, medical, normal*
us_28322	640	480	307200	8	213	RS	*us, medical, normal*
us_28329	640	480	307200	8	213	RS	*us, medical, normal*
us_28348	640	480	307200	8	217	RS	*us, medical, normal*
us_3393	640	476	304640	8	218	RS	*us, medical, normal*
us_3403	584	484	282656	8	256	RS	*us, medical, normal*
us_3405	640	476	304640	8	197	RS	*us, medical, normal*
in_town_09_08_an10	821	536	440056	8	251	JS	*noise*
in_town_09_08_an20	821	536	440056	8	251	JS	*noise*
in_town_09_08_an50	821	536	440056	8	253	JS	*noise*
in_town_09_12_an10	821	536	440056	12	3978	JS	*noise*
in_town_09_12_an20	821	536	440056	12	3972	JS	*noise*
in_town_09_12_an50	821	536	440056	12	4000	JS	*noise*
in_town_09_16_an10	821	536	440056	16	58177	JS	*noise*
in_town_09_16_an20	821	536	440056	16	59358	JS	*noise*
in_town_09_16_an50	821	536	440056	16	59633	JS	*noise*
in_wdrag_09_08	821	536	440056	8	1		*empty*
in_wdrag_09_12	821	536	440056	12	1		*empty*
in_wdrag_09_16	821	536	440056	16	1		*empty*
in_wnoise_09_08	821	536	440056	8	256		*random*
in_wnoise_09_12	821	536	440056	12	4096		*random*
in_wnoise_09_16	821	536	440056	16	65460		*random*

JS) scanned photographs by Dr Jacek Szedel (Silesian University of Technology)
EP) medical images supplied by Prof. Ewa Piętka (Silesian University of Technology)
PH) Philips Medical Systems DICOM Reference Medical Images
ftp://ftp-wjq.philips.com/medical/interoperability/out/Medical_Images/
RS) publicly available DICOM images of the RSNA conference, from devices of various vendors
ftp://wuerlim.wustl.edu/pub/dicom/images/version3/