A COMPARATIVE STUDY OF LOSSLESS CODING TECHNIQUES FOR SCREENED CONTINUOUS-TONE IMAGES

Koen Denecker[1,*]   *Peter De Neve*[2]

ELIS, University of Gent, Sint-Pietersnieuwstraat 41, B-9000 Gent, Belgium
Email: [1]denecker@elis.rug.ac.be   [2]deneve@elis.rug.ac.be

ABSTRACT

The huge sizes of screened colour-separated photographic images makes lossless compression very beneficial for both storage and transmission. Because of the special structure induced by the half-tone dots, the compression results obtained on the CCITT test images might not apply to high-resolution screened images and the default parameters of existing compression algorithms may not be optimal. In this paper we compare the performance of different classes of lossless coders: general-purpose one-dimensional coders, non-adaptive two-dimensional black-and-white coders and adaptive two-dimensional coders. Firstly, experiments on a set of test images screened under different conditions showed that MGBILEVEL and JBIG perform best with respect to compression efficiency; the difference with the other coders is significant. Secondly, we investigated the influence of the screening method (stochastic or classical screening) and screening resolution on the compression ratio for these techniques.

1. INTRODUCTION

The reproduction of continuous-tone images using the four printing inks (CMYK) consists of splitting the image into four colour separations and screening each colour separation under a different angle. New applications, such as printing-on-demand, require an almost identical image to be screened many times; therefore, storing the unchanged part as a screened halftone using lossless compression techniques instead of screening it again and again, may save printing time as well as storage space. Transmitting such images at the very latest moment also benefit from compression. Moreover, the sizes of the images (often about 100 MByte) and the time pressure in the application area (e.g., printing newspapers) make lossless compression even more desirable.

In this paper, we have compared the compression efficiency of several lossless compression schemes. As a reference, we have experimented with four general-purpose byte-oriented coders: GZIP (which is a dictionary-based lossless coder), STAT (an optimized PPM-based compressor by F. Bellard), TIFF PackBits (a simple run-length compressor) and TIFF LZW (Lempel-Ziv-Welch, a dictionary-based compression scheme like GZIP). Then, we have used two nonadaptive coders which accompany the TIFF-standard, namely TIFF Group 3 and Group 4 compression (the former published fax-standards by CCITT [1, 2, 3]). Finally, we have also investigated the compression efficiency (using standard options and after fine-tuning of the parameters)

---

*Research Assistant with the NFWO, Brussels, Belgium

of two two-dimensional black-and-white oriented adaptive coders: BILEVEL coding (by Witten et al.) and JBIG (latest fax-standard).

The compression efficiency of these coders is investigated using six test images, representing halftones at different resolutions, at different screening angles and at different screening techniques.

2. DIGITAL COLOUR SCREENING

The most common procedure in colour screening technology is as follows [4]. Firstly, the CMYK colour image is separated into its four colour channels (cyan, magenta, yellow and black). Each colour separation is a greyscale image, which is then screened under a different angle (e.g., 45° for the most covering colour black, 15° and 75° for cyan and magenta, and 0° for the least visible colour yellow) in order to avoid the appearance of moiré patterns. The screening step itself converts the original greyscale image into dots placed on a rectangular grid. The sizes of the dots are determined by the intensity of the greyscale original. This type of screening is called *classic screening*. Another type of screening, *stochastic screening*, produces equally sized dots at varying distances in order to reproduce the grey tones, giving the dots a random-like appearance.

The screening parameters are illustrated in figure 1. It shows enlargements of the same detail of the cyan separation of the "Musicians" test image screened under different screening methods and screening resolutions. The details are part of the test images in our research.

There are two important parameters for producing digital screens: the screening resolution at which the laser spots are produced, varying from about 300 dpi to about 5000 dpi, and the screen ruling (only in the case of classic screens and expressed in lines per inch), indicating the size of the grid and varying from about 75 lpi for newspapers to 300 lpi for fine art reproductions. Screening a contone image at high resolution increases the image data size drastically, making the need for compression more urgent (e.g., the "Musicians" test CMYK image, 16 Mbyte large, produces 200 Mbyte after halftoning at a fairly high resolution of 2540 dpi).

3. LOSSLESS COMPRESSION SCHEMES

Since each pixel of a screened image is either white or black, a screened image is actually is a bilevel image, albeit with a very special structure. The investigated coders can be divided into three classes: one-dimensional general-purpose coders (GZIP, STAT, TIFF PackBits and TIFF LZW), two-dimensional nonadaptive coders (TIFF Group 3 and Group 4) and two-dimensional adaptive context-based coders (BILEVEL and JBIG).
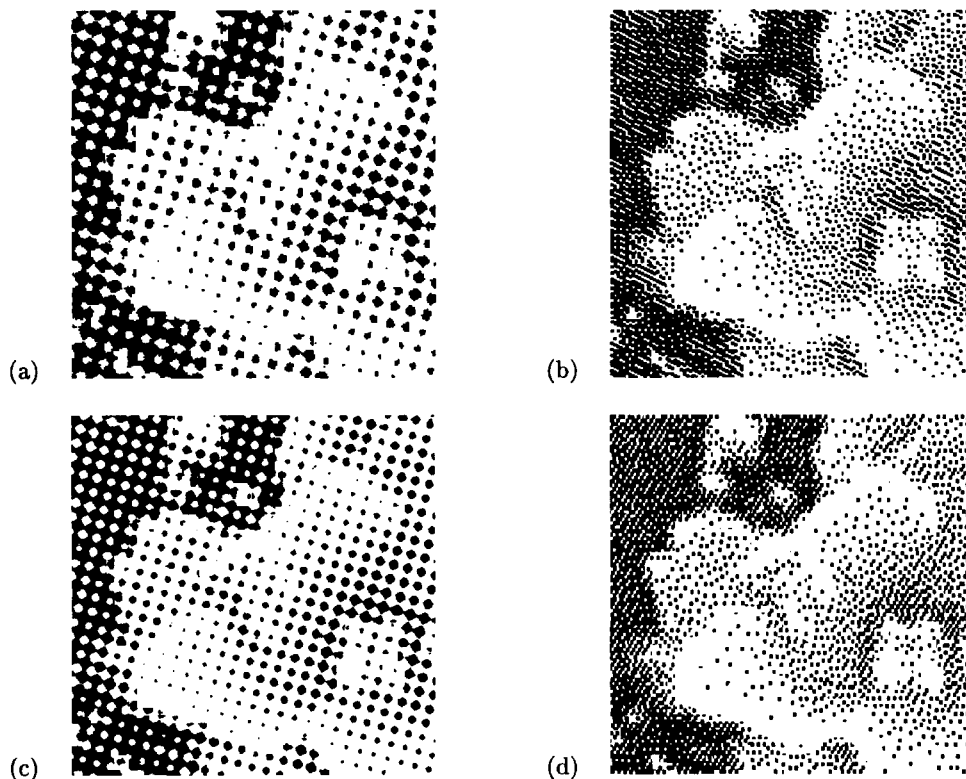
Figure 1. Magnification of the same detail of the "Musicians" image: *(a)* classic screening at 1270 dpi; *(b)* stochastic screening at 1270 dpi; *(c)* classic screening at 2540 dpi and *(d)* stochastic screening at 2540 dpi.

3.1. General-purpose 1-D byte-oriented coders

This first class of coders doesn't exploit the two-dimensional correlation present in images. Neither does it take into account the physical interpretation of the bytes: they are merely interpreted as symbols of an alphabet, disregarding the numerical value associated with these bytes. On the other hand, they are able to adapt themselves one way or another to the information stream.

GZIP (a free and wide-spread compressor by GNU) uses Lempel-Ziv coding (often denoted as LZ77 [5]), which is a form of sliding-window compression. While running through the image file, the window consisting of the last n bytes (e.g., $n = 1024$) is used as a dictionary and each new symbol is replaced by an index to this sliding-window dictionary.

STAT uses a context (i.e., the string consisting of the last 4 symbols) to feed an arithmetic coder [6]. Since it is impossible to construct complete conditional probabilities with a context of 4 bytes, only these specific contexts which arise more than once are taken into account. This approach is called PPM-based, which stands for Prediction by Partial Matching [7].

TIFF Packbits replaces consequent runs of identical symbols by the number of symbols and the value of the symbol; this technique is called run-length coding. It is a very simple and fast algorithm.

TIFF LZW (Lempel-Ziv-Welch) [8] is a dictionary-based technique like GZIP. The technique uses previously coded text as a dictionary. A dictionary consisting of strings is built, in such a way that each shorter string also exists in the library. The encoded files consists of indexes to the longest matching strings.

3.2. Nonadaptive 2-D coders

TIFF Group 3 and TIFF Group 4 are officially known as CCITT Recommendations T.4 [1] and resp. T.6 [2]. These compression schemes are not byte-oriented but treat each line of the image as an alternating sequence of black runs and white runs; only the run-lengths are transmitted using code words defined by the standard, hence the coders are nonadaptive. The code words are chosen to be optimal for scanned textual images, but the nonadaptivity could cause it to perform badly on images with special structures. Group 3 encoding basically is one-dimensional but provides additional options such like two-dimensional encodin and byte-aligning. Group 4 is a simplified or streamlined version of the Group 3 standard in which only two-dimensional coding is allowed.

3.3. Adaptive 2-D context-based coders

These coders use a context, i.e., an amount of previous pixels, to condition the value of each new pixel. Since these coders will prove to be the most efficient ones, we will describe them in more detail than the other.

3.3.1. JBIG

JBIG (Joint Bilevel Image Group) is the newest CCITT lossless facsimile compression standard developed by the Joint Bilevel Image Group [3]. It is an adaptive coder which supports progressive transmission. It uses the Q-coder as a statistical coder but the fact that it's patented by IBM is the most important explanation for the very small popularity.

Organization of the image data The image is separated into resolution layers in case of progressive transmission, bitplanes in case of greyscale images and horizontal

Figure 2. JBIG provides two different types of context templates.

stripes. Coding will take place per layer, per plane and per stripe, called a SDE (Stripe Data Entity).

Contexts A context is defined as a number of neighbouring pixels. Different context templates are chosen for the lowest resolution and the other layers. The lowest resolution template uses 10 pixels, resulting in 1024 different contexts. Two different templates are proposed, as can be seen in figure 2; the left one uses more memory but is more efficient. The current pixel is denoted as "?". Note the existence of an adaptive pixel "A", which can be chosen freely in a much larger neighbourhood of the current pixel. This can be very adequate when compressing periodical structures. The templates at the higher resolution layers use 4 pixels in the lower resolution layer and 6 pixels in the current layer. Two more bits are added to indicate the spatial fase, resulting in 4096 contexts.

Resolution reduction A simple way to produce a lower-resolution image at a quarter of the original size can be achieved by subsampling. Since problems with aliasing or image quality (fine details get lost) can arise, another filter is used, which uses 9 pixels in the current layer and 3 pixels in the lower-resolution layer to produce a new pixel in the lower-resolution layer.

Deterministic prediction A situation can occur that the value of a current pixel can be predicted in a deterministic way based upon the already decoded pixels in the current layer and the values of the pixels in the lower-resolution layer. Then, the value of the current pixel is not coded.

Typical prediction This type of prediction is very useful when large homogeneous regions appear in images. In the lowest-resolution layer, a scanline is said to be typical when it's identical to the previous scanline. In that situation, a special code word is generated instead of coding the entire line and the two corresponding lines in the higher resolution layer are not coded. In the other layers, a pixel is said to be typical when it equals its 8 neighbours as well as the 4 corresponding higher-resolution pixels.

Arithmetic coder The JBIG compressor uses IBM's Q-coder as a statistical coder [9]. This coder is a fast approximation of a binary arithmetic coder, i.e., the symbols can only take on the values of MPS (More Probable Symbol) and LPS (Less Probable Symbol). The code word is defined as the binary representation of the lower limit of the current interval after the entire sequence has been coded. This lower limit only changes when the MPS is encountered and can be achieved by adding the interval corresponding with the LPS. Fixed precision arithmetic is used, which means that a renormalisation must be applied to both code string and interval, at coder and decoder, to maintain the present interval with sufficient accuracy. Carry propagation effects are controlled by bit-stuffing, a zero being inserted when a
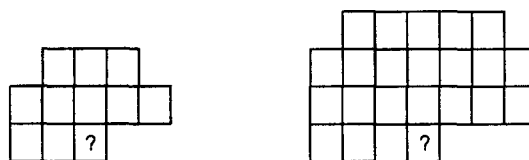

Figure 3. Two context templates are provided: the left one contains 10 pixels and is always used, the right one contains 22 pixels but is only used when necessary.
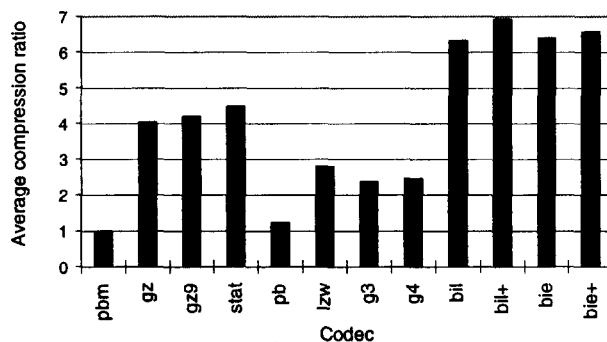

Figure 4. Average compression ratio for the different codecs on the halftone test images (pbm indicates the uncompressed file).

detected run of "1" bits reaches a predefined length. The multiplication is avoided by rounding off probability estimations, which has a minimal effect on coding efficiency.

3.3.2. BILEVEL

BILEVEL is a similar but simpler technique [7] which uses two variable-sized templates as contexts. The smaller context (normally 10 pixels) is always used to condition the new pixel value, the larger context (normally 22 pixels) is only used when it has occurred at least twice, see figure 3. This approach is similar to Prediction by Partial Matching. The two templates can be changed by the user and optimized for a certain application. The advantage of using big contexts is that a better prediction can be obtained; the disadvantage is that it takes much longer to build reliable statistics.

4. EXPERIMENTAL RESULTS

Six test images have been investigated, representing different screening technologies, different screening angles and different screening resolutions. All test images were screened from the same CMYK continuous-tone original "Musicians".

The compression ratios for the different codecs, averaged unweightedly over the set of test images, are shown in figure 4. Some codecs provide options by which they can be optimized for efficiency: "gz9" stands for GZIP -9 and is an optimized but slower version of GZIP, "bil+" is the same as BILEVEL except that the large template has the size of a half diamond and counts 42 pixels instead of 22 pixels and "bie+" means JBIG with several multi-resolution decomposition options altered (a maximal horizontal offset for the adaptive pixel of 16 pixels; 3 differential resolution layers; maximal 20 lines in each stripe in the lowest resolution layer). The options were optimized for one image and are not necessarily optimal for the other halftones.
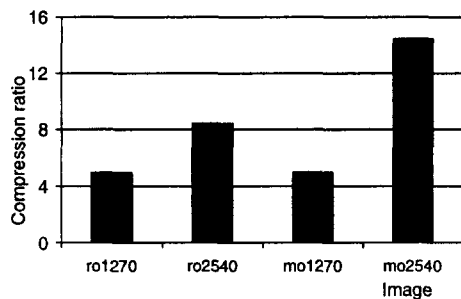
Figure 5. Influence of screening method and screening resolution on compression ratios (only the average of JBIG+ and BILEVEL+ is shown): "ro" indicates classic screening, "mo" indicates stochastic screening, the number indicates the resolution.

From this figure, one can see that the nonadaptive techniques as well as run-length encoding techniques perform worst (average compression ratio less than 3); that the one-dimensional, general-purpose, byte-oriented coders GZIP and STAT perform remarkably well (average compression ratio of 4) and that the best results can be expected from BILEVEL and JBIG (average compression ratio of 7). A larger context size of BILEVEL increases the compression ratio a bit more.

Figure 5 shows the compression ratio, averaged over the two best compressors, as a function of the screening parameters. Combinations of different screening techniques (classic and stocastic screening, indicated by "round dots" and "monet dots" respectively) and different screening resolutions (1270 and 2540 dpi) are compared. The size of the uncompressed file is equal for both techniques (approximately 13 Mbyte at 1270 dpi and 50 Mbyte at 2540 dpi). From this figure, one can see that when doubling the resolution in the case of classic screening, the compression ratio doubles while the umcompressed file increases by a factor of four, which is considered to be normal. But, in the case of stochastic screening, the compression ratio is almost four times as big, which means that the size of the compressed file almost doesn't increase. This leads to the interpretation that the actual image content nearly doesn't increase.

Another conclusion one can draw from this figure is that the compression ratios at normal resolution (1270 dpi) do not depend on the screening technology used, while at higher resolutions, classic screening adds more noise/detail to the screened image than stochastic screening does. This is induced by physical limitations of the ink dot size: though the laser spot resolution gets doubled, the printing dot size remains the same and only the location of the dot can vary.

5. CONCLUSION

In this paper we have compared the compression efficiency of several existing lossless coders on screened continuous-tone images. We have shortly explained the most important screening technologies and the investigated compressors.

The compressors can be divided into three classes: adaptive 1-D general-purpose coders, nonadaptive 2-D coders and adaptive 2-D coders. Experiments with halftones, produced from the same continuous-tone original under different screening conditions, have shown that the adaptive 2-D coders perform best (average compression ratio of about 6), followed by the adaptive 1-D general-purpose coders (average compression of about 4) and the nonadaptive 2-D coders perform worst (average compression ratio of about

2). The difference between the two latest fax-standards TIFF Group 4 and JBIG is significant, unlike for scanned textual documents. Fine-tuning of the parameters of the latter two shows to be useful.

Compression efficiency using these best methods doesn't depend on the screening technique at normal resolutions. However, increasing the resolution leads to normal behaviour of the compression ratio for classic screens, but not for stochastic screens. Because of the physical limitations of the ink dot in the latter case, the actual image content doesn't increase when the screening resolution reaches a certain level.

REFERENCES

[1] The International Telegraph and Telephone Consultative Committee (CCITT), Geneva, *Standardization of Group 3 facsimile apparatus for document transmission, Recommendation T.4*, 1985. Volume VII, Fascicle VII.3, Terminal Equipment and Protocols for Telematic Services. Pages 16–31.

[2] The International Telegraph and Telephone Consultative Committee (CCITT), Geneva, *Facsimile Coding Schemes and Coding Control Functions for Group 4 Facsimile Apparatus, Recommendation T.6*, 1985. Volume VII, Fascicle VII.3, Terminal Equipment and Protocols for Telematic Services. Pages 40–48.

[3] The International Telegraph and Telephone Consultative Committee (CCITT), *Progressive Bi-Level Image Compression. Recommendation T.82*, 1993.

[4] C. Eliezer, "Color screening technology: A tutorial on the basic issues," *The Seybold Report on Desktop Publishing*, vol. 6, October 1991.

[5] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, May 1977.

[6] F. Bellard, "Compression statistique à contexte fini.," June 1995. http://www.polytechnique.fr/~bellard/.

[7] J. G. Cleary and I. H. Witten, "Data compression using adaptive coding and partial string matching," *IEEE Transactions on Communications*, vol. 32, pp. 396–402, April 1984.

[8] T. A. Welch, "A technique for high performance data compression," *IEEE Computer*, vol. 17, pp. 8–19, June 1984.

[9] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon, and R. B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder," *IBM J. Res. Devel.*, vol. 32, pp. 717–726, 1988.