Archive Image Communication with Improved Compression

Xiao Wei Yin, Martin Fleury, and Andy C. Downton
University of Essex
Department of Electronic Systems Engineering
Multimedia Architectures Laboratory
Colchester, Essex, CO4 3SQ, United Kingdom
Tel.: +44 1206 872817
Fax.: +44 1026 872900
{xwyin,fleum,acd}@essex.ac.uk

Abstract

DjVu is a document codec that uses a truncated embedded significance tree to achieve both resolution and image quality scalability. In this paper, the probability model for the truncated tree arithmetic coder is improved, resulting in reduced bit-rates. The trade-off in decoder complexity is also indicated.

1 Introduction

Archive documents are an important resource for research in areas as diverse as climate change, biodiversity monitoring, census statistics and economics, where historical data needs to be compared with modern trends. Whereas recent data is invariably stored in electronic form, historical archives generally require conversion from legacy sources such as card indexes, ledgers or paper files. Central to such conversion is the need accurately to preserve an original document image as well as converting the document content into a suitable electronic database, to preserve parts of the document which may be omitted from the database or erroneously transcribed. In either case, reference to an accurate facsimile of the original document is essential to recover missing information.

Increasingly, document database archives are made universally available via the world-wide web, for which default browser support for JPEG document images is available. However JPEG is not an ideal document image format either in terms of compression efficiency (*i.e.* download time) or reproduction quality, leading to research interest in improved document image coding algorithms. In our work on card archives held at the Natural History Museum in London,[1] preservation of all aspects of document archive cards (including the background) is essential, since critical amendments to the original card data may be made in pencil and lost during OCR conversion.

2. DjVu and Archive Communication

DjVu is a well-regarded document compression tool from AT&T, with algorithms detailed in [5], that gains advantage over JPEG2000 for document compression [3] by separately encoding background and foreground layers [1]. A third bit-map mask layer indicates to each of the other layers which pixels are already coded in the other layer (allowing respective coding coefficients to be discounted [1]). Otherwise, the two codecs are broadly similar: both use a wavelet transform[2]; an embedded significance tree; and a Context Adaptive Binary Arithmetic Coder (CABAC) to encode the significance tree bitstream, which employs bit-plane ordering for progressive coding. The embedded significance tree provides both resolution and SNR (Signal-to-Noise Ratio) scalability. DjVu and JPEG2000 are both more suitable than JPEG for encoding document archives. However, due to their differing design goals, DjVu is significantly more efficient for encoding document images. This paper seeks to further improve DjVu's efficiency in encoding archive documents, bearing in mind the need to preserve accurate facsimiles of both the foreground and background image.

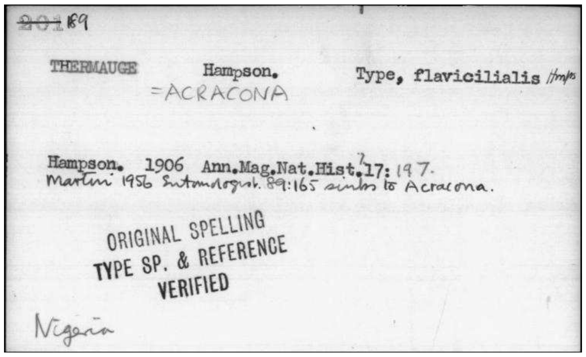
The efficiency of the codec is not the only issue, as memory usage and coder complexity also enter into the comparison. Within DjVu, wavelet coding proceeds on an image block basis (called tiles in JPEG2000). Hence memory usage is considerably reduced, as only a single tile need be held in memory at any one time. Tiling has a limited effect on the overall complexity of the wavelet transform however,

[1]Work was carried out in association with the VIADOCS project, under BBSRC research contracts 84/BIO11933 and 40/BIO11938.

[2]In DjVu, the wavelet codec, IW44, is not applied to the bi-level mask image, for which a variant of the fax. encoder, JBIG2, is applied.

as entropy encoding and quantization can form almost 60% of the total compute time for decoding of high-quality images. Therefore, this paper also considers the practical impact on complexity of varying entropy encoding.

The application for this work arose from (Section 1) a legacy card index, and, therefore, tests took place on cards such as Fig. 1, with segmented background in Fig. 2, produced by the DjVu public-domain segmentation algorithm.



~~201~~ 89

THERMAUGE Hampson. Type, flavicilialis //mp

=ACRACONA

Hampson. 1906 Ann.Mag.Nat.Hist.7 17: 197.

Martin 1956 Entomologist. 89:165 sinks to Acracona.

ORIGINAL SPELLING
TYPE SP. & REFERENCE
VERIFIED

Nigeria

Figure 1. Card showing typical annotations

Figure 2. Background of Fig. 1

3. Improved Entropic Compression

In [10], it is suggested that the DjVu encoder has 'a less efficient embedded representation for each code-block'. Therefore, it is natural to implement extensions from [10] to judge possible improvements. DjVu's IW44 wavelet encoder employs the dyadic Mallat scheme [7] to organize a 2-D image filter bank; successively decimating a LL (Low-pass horizontal, Low-pass vertical) subband to LH (L, High-pass vertical), HL, and HH subbands.[3] 1-D localized wavelet filters are applied in the spatial domain to even and odd row (column) coefficients, or rather their adjacent predictors are interpolated to reduce entropy [9].

[3]The image can be reassembled at increased resolutions by reassembling the corresponding LL, LH, HL, and HH subbands. For simplicity of presentation, this paper assumes a gray-scale image.

However, like JPEG2000 but unlike earlier wavelet encoders operation is on 32×32 image code blocks (in JPEG2000 the block size is larger). Each block employs the dyadic scheme but IW44 has a further decomposition into 4×4 buckets (hence, the algorithm name) with 1, 4, and 16 buckets (called 'precincts' in JPEG2000) according to resolution layer. Each block can, therefore, be randomly selected as a region of interest and formed into a bitstream. In Fig. 3, there are 10 resolution subbands with 0, 1, 2, and 3 the coarsest, and 7, 8, 9 being the finest. An example wavelet coefficient in band four is shown as the parent of the four coefficients in band seven. Selection of coefficients is

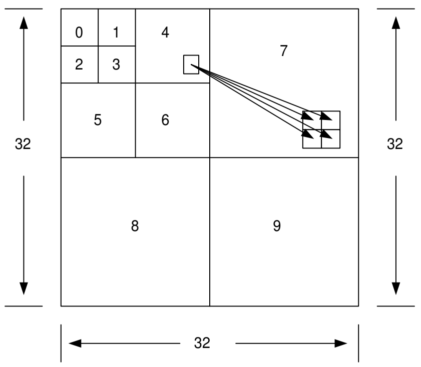


Figure 3. Block decomposition

achieved in IW44 by a generalization [4] of Shapiro's Embedded Zero Wavelet scheme [6] and others, whereby a significance tree is formed. Based on previously decoded information, a decoder can predict whether a transform coefficient is likely to be significant. There are two ways to base a prediction: 1) on values within parent subbands; and 2) on previously coded (in raster scan order) coefficients within the same subband. If one bucket within a subband contains a significant coefficient, then the subband is marked for encoding. Therefore, the significance tree is hierarchical, with the selection scheme in Fig. 3 proceeding from coarse to fine subband (the reverse of the wavelet transform ordering). If an encoding decision is taken then coefficients are bracketed with a given range (though using dead-zone quantization around zero). Successive IW44 slices encode coefficient significance within successive ranges or bit planes, thus allowing the user to control the precision or SNR of the representation by specifying the number of slices.

The significance tree itself is encoded through the IW44 CABAC (the Z'-coder). A binary arithmetic coder [8, 11] outputs a bit if the embedded decision is not apparent or no bit if the probability model accurately predicts the decision. The extent that the previous bitstream code values can accurately predict the next bit is dependant on context variables. For any one decision a relevant context variable acts as an index into a probability look-up-table (LUT) within the statistics module of the coder.[4] Thus, the design of the context model affects the ability to predict, and therefore whether or not a bit needs be output. In the low-complexity Z'-coder [2], there are 98 possible context variables per block arranged in four groups or passes, each pass determining a binary decision: 1) one variable defines whether the buckets within a band should be decoded; 2) 8 variables per 10 bands define whether the coefficients within a bucket should be decoded. 3) 16 variables define whether each coefficient's sign should be decoded and hence its value (scaled by 1.5) calculated from the quantization step size; and 4) one variable decides whether the previous scaling is decreased or increased by 0.5 to renormalize the encoding interval.

4. Supplementing the Context Variables

A number of context variables have been added to the model presented in Section 3. In DjVu (unlike JPEG2000), the significance of coefficients in the same subband is not taken into account. However, as subband data have been observed to have a Laplacian distribution [3], then, with at least one significant coefficient in a bucket, surrounding coefficients are likely to be significant. Pass two and three context variable additions are intended to put back the missing information, though there will be an effect on complexity. The implementation works by incrementing a pointer into a supplemented array of context variables, according to the number of prior neighboring coefficients that are already significant. In turn, the selected context variable indexes the probability LUT. An extra 131 context variables are added in total.

Currently, the decision in pass one on whether a band's buckets should be considered for decoding, solely considers whether it is a root band or its parent was considered active (with coefficients in range), in which case the band is marked as 'new'. In fact, all bands marked as 'new' are automatically considered for bucket decoding in pass two. The first addition (of three context variables) is also to consider whether three or more of the coefficients in the parent band were within range (active), in which case, even though this is a small sample of the available parent coefficients, there is a high probability that the current band has coefficients in range, *i.e.* has decodable buckets.

In pass two, a bucket was only considered for decoding if coefficients in parent buckets were in range. For the higher-layer bands (those with more than four buckets), whether a bucket should be considered for decoding is *now* made conditional on whether neighboring buckets have been marked as 'new'. This doubles the number of context variables for this pass. In Fig. 4 for the finest resolution band with 16 buckets, the relevant previously decoded neighbors for each bucket are indicated. For example, bucket zero has no relevant neighbors, whereas bucket five has three. In pass three,

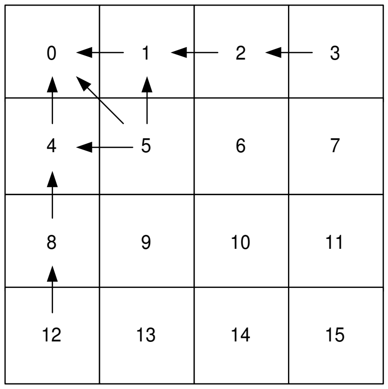


Figure 4. Neighborhood prediction

the decision is equally made depending on whether neighbors are newly in range, according to the same scheme as pass two. Additionally, after [10] context variables (48 in all) are now assigned to the sign of a coefficient (quantization is on the absolute value), whereas in IW44 no variables are assigned to the prediction of the sign. In fact, the sign is likely to be the same as that of neighbors in the horizontal direction, with nearer neighbors being more closely correlated.

4.1. Results

Several test gray-scale images, a sample card, Fig. 1, and and its fore- and back-ground images, Fig. 2, together with the well-known ACM document and Lena images, by way of illustration have been compressed both by means of IW44, and by IW44 with additional context variables, to get a revised file size, as recorded in Table 1. Also included, is a card, Fig. 5, with more than typical penciled-in annotations. For each image, the IW44 file size was progressively increased by selecting a larger number of slices, consequently achieving a larger PSNR (Peak SNR). Though the PSNR is unchanged, the transmitted file size reduces in virtually all cases by the addition of the extra context variables. In other words, for the *same* PSNR the file size improvement is given in Table 1. Comparing for similar file sizes[5], for a typical compression of Fig. 1, file size 10k and 9.9k, for IW44 and revised IW44 respectively, the PSNR improves from 33.50 dB to 34.02 dB, a 1.6% improvement.

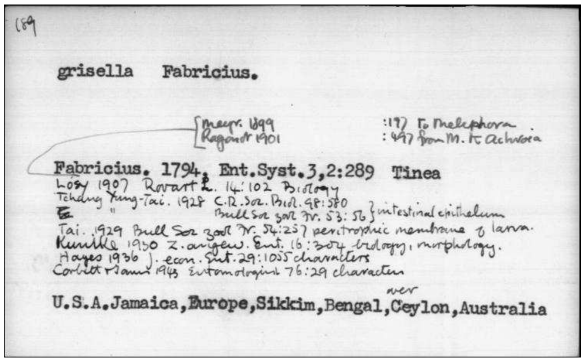
File size reduction is greatest for the smallest number of slices and PSNR (greatest distortion compared to the original image). The technique favors documents over natu-

[4]This arrangement allows the model to be independent from the rest of the coder.

[5]As the unit of DjVu file size quantization is a slice or subband stream, equal file sizes can only be approximated.

IW44 file size	Revised file size	PSNR	Improvement
Complete card image, Fig. 1, 756 × 456 pixels			
5.42k	5.05k	30.59	6.82%
9.40k	8.89k	32.89	5.85%
16.2k	15.6k	37.21	3.70%
36.3k	35.5k	44.38	2.20%
57.6k	56.5k	48.85	1.90%
Foreground of Fig. 1, 756 × 456 pixels			
6.93k	6.58k	28.52	5.05%
11.1k	10.5k	32.56	5.40%
18.7k	18.0k	38.14	3.74%
28.9k	27.9k	44.44	3.46%
39.2k	38.1k	50.70	2.80%
Background of card image, Fig. 2, 756 × 456 pixels			
0.503k	0.445k	36.22	11.50%
0.843k	0.779k	38.81	7.59%
1.60k	1.53k	42.79	4.37%
1.94k	1.87k	43.95	3.60%
3.40k	3.31k	52.44	2.64%
Complete card image, Fig. 5, 756 × 456 pixels			
12.5k	12.0k	29.24	4.00%
26.6k	26.0k	35.20	2.25%
42.4k	41.6k	40.03	1.88%
63.0k	62.1k	44.56	1.42%
90.6k	89.5k	49.35	1.21%
Background of card image for Fig. 5, 756 × 456 pixels			
0.476k	0.424k	35.75	10.90%
0.863k	0.806k	38.64	6.60%
2.19k	2.11k	43.09	3.65%
3.90k	3.81k	50.90	2.30%
4.40k	4.31k	64.46	2.04%
ACM page, 1219 × 1588 pixels			
128k	125k	27.53	2.30%
233k	229k	32.53	1.70%
371k	366k	38.05	1.30%
527k	521k	43.65	1.10%
699k	694k	48.88	0.70%
895k	890k	53.82	0.55%
Lena, 512 × 512 pixels			
3.76k	3.73k	30.28	0.79%
5.95k	5.91k	32.17	0.67%
11.5k	11.5k	35.10	0.00%
22.8k	22.8k	38.16	0.00%
83.8k	84.5k	47.40	-0.83%

Table 1. File size changes from additional context variables



189

grisella Fabricius.

Meyr. 1899 :197 to Melaphora
Ragonot 1901 :497 from M. to Achroia

Fabricius. 1794, Ent.Syst.3,2:289 Tinea
Losy 1907 Rovart L. 14:102 Biology
Tchang Yung-Tai. 1928 C.R. Soc. Biol. 98:580
" Bull Soc zool Fr. 53: 56 } intestinal epithelium
Tai. 1929 Bull Soc zool Fr. 54:257 peritrophic membrane of larva.
Kuntke 1930 Z. angew. Ent. 16:304 biology, morphology.
Hayes 1936 J. econ. Ent. 29:1055 characters
Corbett & Tams 1943 Entomologist 76:29 characters

U.S.A.Jamaica,Europe,Sikkim,Bengal,Ceylon,Australia

Figure 5. Card with considerable annotations

ral images such as Lena, which contain high-frequency elements such as those around the hair. Smooth document images such as the card background fare better than images with high-spatial frequency content. The improvement on the foreground only is consistent with that on the transformed card image. Comparing, the result for Fig. 1 with that for Fig. 5, if there are considerable annotations then the gain to the complete/foreground image is likely to be small. Further tests (for reasons of space not reported herein) on a variety of card images demonstrated similar results. Card backgrounds results are broadly similar or improved upon the samples, which indicates the principal advantage of this enhancement. The effect of each of the changes in Section 4 was measured in Tables 2 & 3 for respectively Figs. 1 & 2. It became evident that the change to pass two does not always yield results, and that the change to pass three alone can result in an increase in file size. However, taken with the sign encoding in pass three, the change to pass three gives an improvement, and the net result is always an improvement in output file size (bit-rate).

Table 4 records combined quantization and CABAC timings on a Pentium 4 at 1.7 GHz for Fig. 1 before and after the changes. Also included are timings for the complete decoder. Timings were made with the supplied DjVu internal timer routines. Taking the partial results alone, there is about a 25% time overhead for lower PSNR images, but this decreases to around 10% for higher PSNR images.

5. Conclusion

The probability model for the binary arithmetic coder responsible for DjVu progressive coding has been improved. An enhanced set of context variables results in up to 10% improvement in the bitstream size, depending on SNR resolution. There is a complexity trade-off that must be balanced against the reduced transmission time. These trade-offs are important for Internet access of legacy archives such as the card index application considered herein. For this class of images, the DjVu background layer is generally smooth, and gains most from the additions, and for most foregrounds the effect is similar, though with increasing handwritten modifications to the printed card, the gain is reduced. Knowledge of the different markings present on the cards, such as pencil, ballpoint, or ink pen, may allow other layers, as well as the foreground layer, to be extracted. Research is also in progress to find savings from considering a sequence of card images or other documents as a whole, rather than coding each card separately. 3D wavelet coding has already been investigated, and the intention is to examine combined wavelet transform with Karhunen-Loève transform or vector quantization in the document sequence dimension.

References

[1] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, and Y. LeCun. High quality document image compression with DjVu. *Journal of Electronic Imaging*, 7(3):410–425, 1998.

[2] L. Bottou, P. G. Howard, and Y. Bengio. The z-coder adaptive binary coder. In *Data Compression Conference*, 1998.

[3] C. Christoploulos, A. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system: An overview. *IEEE Transactions on Consumer Electronics*, 46(4):1103–1127, 2000.

[4] G. M. Davis and S. Chawla. Image coding using optimized significance tree quantization. In J. A. Storey and M. Cohn, editors, *Data Compression Conference – Designs, Codes, & Cryptography*, pages 387–396, 1997.

[5] Specification of DjVu image compression format. Technical report, AT & T, 1999. Report 1999-29-04 15:46 EDT available from LizardTech Inc., Seattle, WA.

[6] M. Ghanbari. *Video Coding: An Introduction to Standard Codecs*. IEE, London, 1999.

[7] S. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 11:674–693, July 1989.

[8] W. B. Pennebaker, J. L. Mitchell, G. G. Langdon Jr., and R. B. Arps. An overview of the basic principles of the Q-Coder adaptive binary arithmetic coder. *IBM Journal of Research and Development*, 32(6):717–726, 1988.

[9] W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Journal of Applied Computing and Harmonic Aanalysis*, 3(186-200), 1996.

[10] D. Taubman. High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, 9(7):1158–1170, July 2000.

[11] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan-Kaufmann, San Franisco, $2^{\underline{nd}}$ edition, 1999.

IW44 file size	Change to	Revised file size
5.42k	Pass 1 only	5.38k
	Pass 2 only	5.45k
	Pass 3 only	5.29k
	Sign encode only	5.29k
	Pass 3 and sign encode	5.12k
	All	5.05k
36.3k	Pass 1 only	36.2k
	Pass 2 only	36.3k
	Pass 3 only	35.9k
	Sign encode only	36.3k
	Pass 3 and sign encode	35.7k
	All	35.5k

Table 2. Detailed effect on Fig. 1

IW44 file size	Change to	Revised file size
0.503k	Pass 1 only	0.496k
	Pass 2 only	0.501k
	Pass 3 only	0.500k
	Sign encode only	0.464k
	Pass 3 and sign encode	0.450k
	All	0.445k
1.94k	Pass 1 only	1.91k
	Pass 2 only	1.94k
	Pass 3 only	1.93k
	Sign encode only	1.91k
	Pass 3 and sign encode	1.89k
	All	1.87k

Table 3. Detailed effect on Fig. 2

IW44 file size	Revised file size	IW44 decode partial (full) time (ms)	Revised decode partial (full) time (ms)	Increase in time %
5.42k	5.05k	30 (80)	40 (90)	33 (12)
9.40k	8.89k	40 (90)	50 (110)	25 (22)
16.2k	15.6k	80 (130)	90 (140)	12 (7)
36.3k	35.5k	140 (200)	160 (230)	14 (15)
57.6k	56.5k	220 (280)	240 (300)	9 (7)

Table 4. Partial (full) decoder timings for Fig 1, excluding disc I/O