Can the sample being transmitted be used to refine its own PDF estimate?

Dinei A. Florêncio and Patrice Simard
Microsoft Research
One Microsoft Way, Redmond, WA 98052
{dinei, patrice}@microsoft.com

Abstract

Many image coders map the input image to a transform domain, and encode coefficients in that domain. Usually, encoders use previously transmitted samples to help estimate the probabilities for the next sample to be encoded. This backward probability estimation provides a better PDF estimation, without need to send any side information. In this paper we propose a new method of encoding that goes one step further: besides past samples, it also uses information about the *current* sample in computing the PDF for the current sample. Yet, no side information is transmitted. The initial PDF estimate is based on a Tarp filter, but probabilities are then progressively refined for non-zero samples. Results are superior to JPEG2000, and to bit-plane Tarp.

1. Introduction

Many image coders [1],[2],[3],[4] are based on mapping the input image into a transform domain, quantizing the samples in that domain, and sending the quantized samples to the decoder, using some type of entropy coding. The wavelet transform is particularly popular as the choice of transform, and will be used in this paper. Figure 1 illustrates a typical wavelet encoder. The wavelet coefficients are quantized (divided by a quantization step Q and then rounded to nearest integers), and the resulting indices are encoded without loss by an entropy encoder. The entropy coder typically uses a probability distribution function (PDF) over the quantized values, and the number of bits it produces is close (typically within less than 1%) to the entropy of the PDF. The PDF is computed adaptively from previously encoded coefficients (kept in the "store" box), by counting frequencies for each context, for example. Higher compression is obtained by increasing Q, which decreases the entropy of the PDF, mostly because more coefficients quantize to zero.

Performance of the coder is dependent essentially on the spatial transform (e.g., wavelet), and on the quality of the PDF estimates provided by the "Adaptive PDF" box. In general, the "current pixel" is not used in computing the PDF, since it is not available at the decoder. This is one of the main contributions of this paper: we show that, in a sense, it is possible to exploit the current sample to enhance the quality of the estimated PDF. For that purpose, we use partial information about the current sample to refine the estimate of its own PDF.

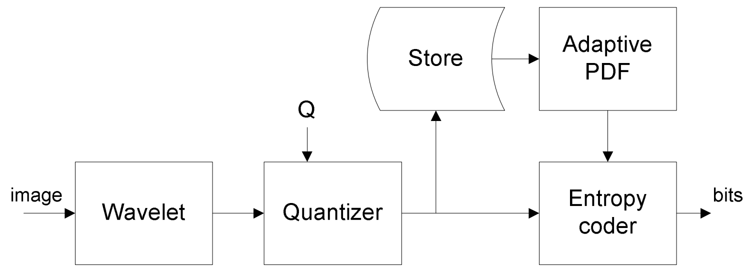

Figure 1. Simplified block diagram of a wavelet-based signal coder.

The basic idea is to transmit the information about the current sample in parts. More precisely, we first send the info regarding the current sample being zero or not. If it is zero, we are done and go to the next sample. If not, we then re-estimate the PDF to take into account the newly received info that the sample is not zero. We propose a simple way of doing this without exploding the computational complexity requirements.

In the next section we review the basics of partial information transmittal. In Section 3 we review the Tarp filter and present our preliminary PDF estimator, which is based on a Tarp filter. In section 4 we present the details of the PDF estimation refinement, and its coupling with the arithmetic encoding process. In section 5 we present some experimental results, which show the superiority of the proposed algorithm to JPEG2000 and bitplane Tarp, on the Kodak reference set. Conclusions and final comments are presented in Section 6.

2. Partial Information Transmission

Traditionally, we estimate a PDF based on the information already available at the decoder (to avoid side-information), and some pre-accorded signal model, of which both encoder and decoder are aware. For example, one may assume the wavelet coefficients have a zero-mean Laplacian PDF, and estimate the variance based on the previously transmitted samples.

One interesting result of information theory is that partial information transmission is free. More precisely, if $A \Rightarrow B$, then $p(B|A)=1$, and we can write:

$$p(A) = p(B \mid A).p(A) = p(A, B) = p(A \mid B).p(B) \tag{1}$$

and therefore:

$$\mathrm{I(A) = I(A|B)+I(B)}, \tag{2}$$

where $I(A) \triangleq \log_2(p(A))$ is the amount of information in A. For example, assume x $\in$ {1, 2, 3, 4}, and has a PDF p(x) = {.5, .25, .125, .125}. The cost of directly sending the information x=4 is 3 bits. But the cost of sending the sequence {x>1, x>2, x>3} is the same, as long as all the information already transmitted is used to condition the subsequent probabilities. In the example, the cost of sending x>1 is one bit, since p{x>1} = .5. The cost of sending {x>2|x>1} is also one bit, and finally, the cost of sending {x>3|x>2} is also one bit. Note that the cost of {x=4|x>3} is zero bits, since that's the only possibility for x, given that x>3. This result illustrated in this example is independent of the PDF, or the particular alphabet, as long as the entropy coder is perfect. In summary, and since arithmetic coding is nearly perfect, for all practical purposes partial information transmission is essentially free.

3. The Tarp Filter

The Tarp filter was introduced in [2] as a simple way of directly estimating the binary probabilities of each bit in a bit plane. With excellent performance in a simple structure, the Tarp encoder challenges the traditional notion that context needs to capture spatial structure of the surrounding pixels. Being an IIR filter, the Tarp filter has the nice property of producing a long impulse response with low computational requirements. In [2] the Tarp filter was applied directly to the neighboring bits to produce an estimate of the probability of the current bit of each wavelet coefficient. Recent results by other researchers validate these surprising results [5].

In this paper we use a Tarp filter to help estimate the probabilities of multilevel wavelet coefficients (not in a bitplane fashion). In order to do that, we estimate the variance of each wavelet coefficient, based on its neighbors. We also use another Tarp filter to obtain a variance estimate based on the previous band, which helps in providing information about the region of the image not yet scanned in the current band. A PDF is then obtained based on that variance and an underlying model. In this section we give some details of our use of the Tarp filter, and, in Section 4 we will explain how we use this variance estimate to obtain our progressively refined PDF estimate.

3.1. A simple 1-D filter for variance estimation

To help understand the Tarp filter, we start by considering a one-dimensional scenario. We can build an estimate of the variance of the symbol to be encoded by using a simple first-order recursive filter:

$$\sigma[t] = a\sigma[t-1] + (1-a)\left(v[t-1]\right)^2, \tag{3}$$

where σ[t] is the estimate of variance for sample v[t], v[t] is the value of the wavelet coefficient at position t, and *a* is a learning rate parameter between 0 and 1, which controls how quickly the probability estimate adapts to the data.

It is easy to show that $\sigma[t]$ is the convolution of $(v[t])^2$ with the filter impulse response $f[t] = a^t(1-a)$ for $t \geq 0$, and 0 otherwise. The parameter *a* can also be viewed as a smoothing factor; the noisier the data, the higher we should set the value of *a*. The main advantage of this algorithm is simplicity, since very few operations are involved.

In [2], the Tarp filter was used to directly compute the probability of a given bit being zero (or one). Here, we assume the wavelet coefficients have a certain PDF, say a Laplacian distribution, and use the tarp filter simply to estimate the variance of the Laplacian for each wavelet sample.

3.2. Combining four 1-D filters to create the Tarp 2-D filter

If we generalize the simple filter discussed before to 2-D, scanning order becomes an issue. As in [2], we simply use the usual raster scanning for images, line-by-line, from left to right, trying to extract the maximum information from all previously seen pixels (i.e. located in a line above or located to the left of the current pixel). This is done by using four 1-D filtering steps. The first filter goes from left to right and is similar to the 1-D filter described above. The second filter goes from right to left, and is done after each full line has been processed. The resulting probabilities are kept in a buffer. The third filter goes from top to bottom for each column, using the probability computed in the previous line. Finally, a fourth filtering step is performed after the band is completely encoded, and will be used to encode/decode the next band of the same type (e.g., HL, LH, or HH). Differently from the Tarp filter described in [2], we use a different learning parameter for the top to bottom (i.e., vertical) process. More precisely, we use a_H=0.25 for the horizontal filtering and a_V=0.5 for the vertical filtering, respectively. Also, to keep the symmetry of the process, the HL Wavelet band is rotated by 90 degrees before encoding.

The computation of the Tarp filter is summarized by two sets of equations. The first set implements the four 1-D filters:

$$\sigma_{H1}[i,j] = a_H \sigma_{H1}[i-1,j] + (1-a_H)(v[i-1,j])^2 \tag{4}$$

$$\sigma_{H2}[i,j] = a_H \sigma_{H2}[i+1,j] + (1-a_H)(v[i,j])^2 \tag{5}$$

$$\sigma_{V1}[i,j] = a_V \sigma_{V1}[i,j-1] + \left(\frac{1-a_V}{1+a_H}\right)(a_H \sigma_{H1}[i,j-1] + \sigma_{H2}[i,j-1]) \tag{6}$$

$$\sigma_{V2}[i,j] = a_V \sigma_{V2}[i,j+1] + \left(\frac{1-a_V}{1+a_H}\right)(a_H \sigma_{H1}[i,j] + \sigma_{H2}[i,j]) \tag{7}$$

where σ_{H1}, σ_{H2}, σ_{V1}, and σ_{V2} are the recursive estimates coming from the left, right, top and bottom, respectively.

Note that σ_{H2} and σ_{V2} include non-causal samples, and therefore we must combine them in such way that guarantees causality. The filter σ_{V1} is causal because it depends on σ_{H2} of the previously transmitted line. We first note that all four equations can be used if we refer to a lower resolution band, which has already been transmitted. Therefore, a causal variance estimate can be obtained as:

$$\sigma[i, j] = w_C \sigma_{CB}[i, j] + (1 - w_C)\sigma_{PB}[i/2, j/2], \tag{8}$$

where, w_C is the relative weight for the variance estimate for the current band (we used w_C=0.875 in our experiments), σ_{CB} is the variance estimate based on the current band, and σ_{PB} is the variance estimate based on the previous band of same nature (e.g., HL, LH, or HH), but at lower resolution (when available). Due to the progressive transmission across wavelet bands, σ_{PB} is always causal because it is computed after the whole band has been transmitted. It can be computed as:

$$\sigma_{PB}[i, j] = \left(\frac{1}{1 + a_V} \right) \left(a_V \sigma_{V1}[i, j] + \sigma_{V2}[i, j] \right) \tag{9}$$

The filter σ_{CB} is causal since we have already established that σ_{H1} and σ_{V1} are causal:

$$\sigma_{CB}[i, j] = w_V \sigma_{V1}[i, j] + (1 - w_V)\sigma_{H1}[i, j] \tag{10}$$

where w_V is a weighting factor, which we set experimentally to 0.4.

In all the computations, initial conditions at the boundaries are given by an a priori estimate of the variance of v[i,j].

4. PDF estimation.

Our encoder consists basically of a wavelet transform followed by a (non-adaptive) arithmetic encoder. Therefore, the coding efficiency will depend heavily on the quality of the PDF estimate provided to the arithmetic encoder. In other words, our basic task is to provide the arithmetic encoder with a good PDF estimate of the wavelet coefficients. A typical approach is to divide the samples in classes and let the arithmetic encoder adaptively estimate a PDF for each of these classes. These classes are generally based on the surrounding pixels that have already been transmitted, and therefore may include both variance and accurate positional information. For instance, the context could contain an edge, and the PDF can depend on where the current pixel lands with respect to this edge. One disadvantage of this method is that it may take a long time for the arithmetic encoder to build up significant data to reliably predict the PDF. This implies that the statistics have to be built over an extensive area of the image, with severe implications in robustness to transmission errors. Instead, we follow here a much less data intensive path: we assume the PDF is Laplacian, and simply estimate the variance based on the previously transmitted samples. If robustness to errors is desired, one can easily reset the estimate at image blocks, therefore limiting the spreading of any eventual transmission errors. Furthermore, we do not transmit a wavelet coefficient in a single step. Instead, we use the partial information transmission principle (described in Section 2) to progressively transmit each coefficient. And we refine our PDF estimate in between each partial information transmission.

Figure 2 illustrates the basic procedure of encoding a wavelet coefficient. First, an estimate of the variance is done, based on a Tarp filter, as described in the next Section. We assume the PDF is a zero-mean Laplacian, i.e.,

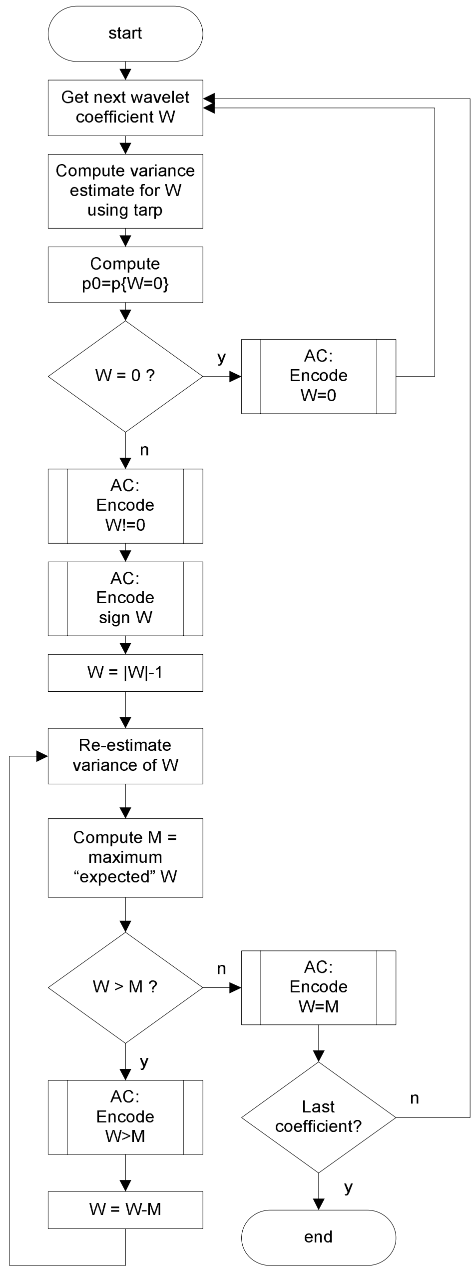

Figure 2. Simplified block diagram of the encoding of one wavelet coefficient.

$$p(a) = \frac{1}{\sqrt{2}\,\sigma} e^{-\frac{\sqrt{2}}{\sigma}|a|}. \qquad (11)$$

where σ is the variance of the coefficient, after normalization by the quantization step. The first part of the progressive transmission is to send whether the quantized wavelet sample is zero or not. Since we use a uniform quantizer, the probability P(0) of the quantized value q(a)=0 can be found by integrating (11) between -.5 and +.5, and is:

$$P(0) = 1 - e^{-\frac{1}{\sqrt{2}\sigma}} \qquad (12)$$

This probability of the coefficient being zero is computed, and used by an arithmetic encoder to encode this zero/non-zero information. If the coefficient is zero, we are done, and can proceed to the next sample. If not, we need to send the sign of the coefficient, as well as the value itself. After transmitting the sign, we can take the absolute value of the sample, and the Laplacian distribution becomes an exponential distribution. We can now encode the non-zero coefficient by using this exponential distribution. Nevertheless, before sending the actual value of the coefficient, we will use the information that the coefficient is not zero to re-estimate the coefficient variance. In particular, we use a simple update, just to eliminate the cases where the initial variance estimate was excessively low:

$$\sigma' = \begin{cases} \sigma & if \quad \sigma > 0.4 \\ 0.4 & otherwise \end{cases} \qquad (13)$$

The value 0.4 was obtained experimentally, but did not seem to be particularly sensitive, and any value between 0.2 and 1 yielded similar results.

Before sending the next symbol, we also estimate a maximum "expected" value. This has two objectives: First, it allows us another opportunity to re-estimate the variance, if the value is bigger than this maximum value. Second, it allows keeping the required precision of the arithmetic encoder under control. More precisely, higher values have lower probability, since we assume an exponential distribution for the non-zero samples. Depending on the estimated variance, and the particular wavelet coefficient, the probability of a certain value may become lower than the precision allowed by a finite precision arithmetic encoder. We limit the maximum value for this partial information transmission at:

$$\text{Max\_Coef} = \text{floor}\,(-\,\sigma/\sqrt{2}.\log(\text{MIN\_PROB}/(1\text{-}e^{-\sqrt{2}/\sigma}))); \qquad (14)$$

where we used MIN_PROB = 2^{-11}, so that the minimum probability for a symbol is at least 8 times higher than the probability precision used in our arithmetic encoder. In our experiments, we observed that the particular choice of MIN_PROB has negligible effect in the compression performance, but it allows to keep under control the precision required by the arithmetic encoder.

The probabilities of all symbols higher than Max_Coef are grouped into a single symbol (i.e., an "escape code"). If an escape code is sent, than the variance is again re-estimated, and the symbol is re-encoded. If the coefficient is still too large, the process may be iterated as many times as needed.

The Tarp filter has low computational requirements and the proposed recursive PDF estimation and arithmetic encoding can be done in an efficient way. The estimated PDF is always an exponential, which means we can directly compute the cumulative probabilities for the current symbol, as required by the arithmetic encoder. In other words, we do not need to compute probabilities for each and every symbol. A careful implementation exploiting these properties of the encoder should have very low computational requirement.

5. Experimental Results

We have implemented the proposed encoder and evaluated its performance with the gray scale images from the same Kodak grayscale 768x512 image set used in [3], and compare the results to those of JPEG, PWC [3], bitplane Tarp [2], and JPEG2000. For all codecs, intermediate bitstream files were generated and decoded, so the compressed file size included bookkeeping and format overheads. For each image, the usual 7-9 biorthogonal wavelet transform is computed, with 5 subband levels. Furthermore, since we do not use context adaptive probability estimation, we do intra-band wavelet prediction. Each wavelet coefficient is predicted as

$$v'[i,j] = v[i,j] - \beta_H \hat{v}[i-1,j] - \beta_V \hat{v}[i,j-1] \quad (15)$$

where β_H and β_V depend on the wavelet band, and are +.125, and -.125 for the HL bands, -.125, and +.125 for the LH bands, and -.125, and -.125 for the HH bands.

The variance estimate produced by the Tarp filters is then used by the (non-adaptive) arithmetic encoder to encode each wavelet coefficient, except that the variance may be re-estimated in the process, as explained in Section 4. For each non-zero symbol, the sign is coded independently in raw mode, i.e. without any compression. Note that this could be improved by using sign prediction [5].

Image	JPEG	PWC [3]	JPEG2000	Tarp [2]	PT-Tarp
1	2.64	2.52	2.45	2.40	2.21
2	1.27	1.04	1.06	0.98	0.94
3	0.75	0.62	0.57	0.56	0.54
4	1.30	1.05	1.06	0.98	0.96
5	2.50	2.33	2.24	2.19	2.00
6	1.92	1.74	1.69	1.62	1.54
7	0.94	0.78	0.72	0.72	0.67
8	2.80	2.58	2.45	2.45	2.29
9	0.90	0.75	0.68	0.68	0.65
10	1.06	0.87	0.81	0.80	0.75
11	1.75	1.55	1.50	1.45	1.36
12	1.03	0.85	0.79	0.78	0.75
13	3.32	3.15	3.17	3.01	2.80
14	2.26	2.01	2.00	1.89	1.76
15	1.10	0.93	0.89	0.85	0.83
16	1.35	1.14	1.11	1.06	1.02
17	1.25	1.02	0.99	0.95	0.90
18	2.26	2.02	2.02	1.91	1.84
19	1.64	1.43	1.39	1.33	1.30
20	0.93	0.78	0.72	0.72	0.71
21	1.63	1.44	1.42	1.35	1.30
22	1.69	1.49	1.50	1.40	1.35
23	0.56	0.38	0.37	0.35	0.36
24	2.02	1.87	1.80	1.76	1.66
Average	**1.62**	**1.43**	**1.39**	**1.34**	**1.27**

Table 2. Compression performance, in bits per pixel, of several grayscale image codecs, for a PSNR of 40 dB.

A typical set of results for the image set are show in Table 2, for a peak signal-to-noise ratio (PSNR) of 40.0 dB (which leads to high visual quality, with essentially imperceptible quantization artifacts for most images). Similar results are obtained at other PSNR settings..

We note that the Tarp-based codecs had the best compression performance for that group of codecs, in the Kodak image set. The average improvement over JPEG2000 is over 8%, putting in further evidence of the effectiveness of the PDF estimation without using context information. The improvement of around 5% over the original Tarp encoder [2] illustrates the effectiveness of the progressive information transmission approach when combined with multilevel wavelet encoding.

6. Conclusion

In [2] a very simple codec based on the Tarp filter was introduced. The probabilities of each bit were obtained directly from the previously transmitted bits and used to control a non-adaptive arithmetic coder. This result was important because it provides a new interpretation of the relative importance of contextual information, since the Tarp filter does not incorporate accurate positional information (e.g. edge estimation) into the probability estimate. Because the Tarp filter in [2] operates directly on the bits, a new way of using it is necessary when encoding multilevel wavelet coefficients without using bitplanes. In this paper we cover that gap: we introduced a way of using a 2-D “Tarp” filter, to obtain probability estimates for *multilevel* wavelet coefficients. Instead of directly computing probabilities, we use the Tarp filter to predict the variance of the coefficient, and then use that to obtain a PDF estimate by using a PDF model. We have also shown how we can refine the PDF estimate during the transmission of a coefficient by using the partial information transmission principle.

The Tarp filter has low computational requirements and the proposed recursive PDF estimation and arithmetic encoding can be done in an efficient way.

Like the original Tarp, our encoder does not use of accurate positional information. The performance – around 8% better than JPEG2000 – is similar to much more complex encoders, which make use of this information. We believe incorporating this information into a Tarp encoder will yield even better results

References

[1] P. Simard and H. S. Malvar, “A wavelet coder for masked images,” *Proc. IEEE Data Compression Conf.*, Snowbird, UT, pp.93–102, Mar. 2001.

[2] P. Simard, D. Steinkraus, and H. Malvar, “On-Line adaptation in image coding with a 2-D Tarp filter,” *Proc. IEEE Data Compression Conf.*, Snowbird, UT, pp. 23-32, Mar. 2002.

[3] H. S. Malvar, “Fast progressive wavelet coding,” *Proc. IEEE Data Compression Conf.*, Snowbird, UT, pp. 336–343, Mar. 1999.

[4] S. LoPresto, K. Ramchandran, and M. T. Orchard, “Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework,” in *Proc. IEEE Data Compression Conf.*, Snowbird, UT, pp. 221–230, Mar. 1997.

[5] James Fowler, Personal communication. Independent implementation of Tarp available in QccPack (http://qccpack.sourceforge.net/).

[6] A. Deever and S. S. Hemami, “Efficient sign coding and estimation of zero-quantized coefficients in embedded wavelet image codecs,” *IEEE Trans. on Image Processing*, accepted for publication.

[7] A. Mo at, R. M. Neal, and I. H. Witten, “Arithmetic coding revisited,” *ACM Trans. Information Systems*, vol. 16, pp. 256–294, 1998.

[8] H. S. Malvar, “Fast adaptive encoder for bi-level images,” *Proc. IEEE Data Compression Conf.*, Snowbird, UT, Mar. 2001.