SHAPE-ADAPTIVE TARP CODING

James E. Fowler

*Department of Electrical and Computer Engineering
Engineering Research Center
Mississippi State University, Mississippi State, MS*

ABSTRACT

An embedded wavelet coder is proposed for arbitrarily shaped image objects. Contrary to the popular approach of considering transparent regions surrounding image objects as permanently insignificant, the proposed coder imposes no assumptions on the significance state of unknown transparent regions. Instead, the proposed shape-adaptive tarp coder employs a simple algorithm called tarp filtering to produce a probability estimate of coefficient significance which propagates unchanged across transparent regions. Experimental results show that the shape-adaptive tarp coder offers coding performance superior to a popular zerotree-based technique and equivalent to a popular method based on context conditioning.

1. INTRODUCTION

A common approach to handling arbitrarily shaped image objects within modern embedded wavelet coders is to employ a shape-adaptive discrete wavelet transform (SA-DWT) [1] to transform only non-transparent regions in the image, and to permanently consider transparent regions in the image as “insignificant” coefficients. The encoding algorithm then processes these transparent regions in a manner identical to that of other insignificant coefficients. This approach has been taken for a number of zerotree-based coding algorithms in which zerotree structures aggregate large regions of transparent coefficients into zerotree symbols along with the opaque insignificant coefficients. A further refinement on this basic approach observes that greater efficiency can be obtained by discarding all sets of coefficients that lie entirely within a transparent region from consideration by a zerotree-based coder [2]. This refinement typically offers a small gain in performance, although the size of this gain is dependent on how much of the overall image is transparent. Treating transparent regions as insignificant coefficients can also be applied to create shape-adaptive image coders from techniques that rely on context conditioning rather than zerotrees. Although the most prominent example of such techniques, JPEG-2000, does not support arbitrarily shaped image coding, the underlying EBCOT algorithm is easily made shape-adaptive. In shape-adaptive EBCOT [3], transparent regions are ignored and not coded in all coding passes, while anytime that the context for an opaque coefficient overlaps the object boundary, transparent coefficients in the context are treated as insignificant.

By definition, there is no data associated with transparent regions of an object; consequently, the significance or insignificance of these transparent regions cannot be known. The drawback to the approach of permanently marking transparent regions as insignificant is that this imposes an assumption on the significance

state of unknown regions, and the coder is forced to incorporate this artificial data in the coding of opaque regions, either indirectly if the form of context conditioning, or directly in the form of tree structures that overlap object boundaries. Often, the assumption imposed (i.e., insignificance) bears little resemblance to the data in the opaque regions that is actually known. For example, even if half of the coefficients in the opaque regions are known to be significant, the transparent coefficients are always constrained to be insignificant. Of course, in many situations, the effects of this constraint may be negligible since the coding of opaque coefficients only near the object boundaries are affected by the imposition of insignificance in the transparent regions. However, if the transparency mask is particularly complex, with a relatively large proportion of opaque coefficients lying on or near the object boundaries, the effect may no longer rest non-negligible. In these cases, better rate-distortion performance may be obtained by avoiding the imposition of assumptions on the significance state of unknown transparent regions.

In this paper, we propose a shape-adaptive coder that does not assume any information on the significance state of transparent regions. Our coder is a modification of the tarp coder proposed by Simard *et al.* [4]. The tarp algorithm involves a progressive “bitplane” coding of wavelet coefficients using a Parzen-window technique to estimate probability of significance and nonadaptive arithmetic coding to code significance-map and refinement information. As proposed originally by Simard *et al.* as a full-frame (i.e., not shape-adaptive) coder, the tarp coder offers rate-distortion performance surprising close to that of other, more sophisticated coders despite the fact that the tarp coder lacks complex context modeling or cross-subband, cross-scale aggregation of symbols such as zerotree structures. Our modification to the tarp coder to handle shape-adaptive coding calls for “skipping” over the transparent regions while maintaining the current probability estimate unchanged. Because the tarp algorithm lacks context modeling and symbol aggregation, this skipping of transparency leads to efficient performance for shape-adaptive coding.

Below, we describe our shape-adaptive tarp coder, first reviewing the theory behind Parzen windows that drives the tarp algorithm and then briefly describing the tarp coder as presented in [4]. We then describe our shape-adaptive tarp algorithm and compare its performance to both a zerotree coder (shape-adaptive SPIHT [2]) as well as a context-based coder (shape-adaptive EBCOT [3]).

2. SIGNIFICANCE CODING VIA PARZEN WINDOWS

Consider a 2D field of real-valued coefficients, $c(x, y) \in \mathbb{R}$, where x and y are integer-valued. Given a threshold $t \in \mathbb{R}$, the coefficient at location (x, y) is significant if $|c(x, y)| \geq t$, and is insignificant otherwise. Define the *significance state* with respect to t of $c(x, y)$

This work was funded in part by the National Science Foundation under the Large Data and Scientific Software Visualization Program, Grant No. ACI-9982344.

to be

$$v(x, y) = \begin{cases} 1, & |c(x, y)| \geq t, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Suppose we know that coefficients at locations $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ are significant with respect to some given threshold, and we would like to estimate the probability that the coefficient at location (x, y) is also significant. Parzen windows [5] is one approach to performing this probability estimate. Specifically, we estimate the probability that $c(x, y)$ is significant as

$$p(x, y) = \sum_{i=1}^m \phi(x - x_i, y - y_i), \quad (2)$$

where $\phi(x, y)$ is a 2D window sequence. A possible window sequence which is suited to the well known Laplacian distribution of wavelet-coefficient magnitudes in images is the Laplacian window,

$$\phi(x, y) = \beta \alpha^{|x|+|y|}, \quad (x, y) \in \mathcal{R}, \quad (3)$$

where α is a parameter controlling the spread of the window, and β is chosen so that

$$\sum_{(x, y) \in \mathcal{R}} \phi(x, y) = 1, \quad (4)$$

over the region of support of the window, $\mathcal{R}$. As a result, it can be shown [5] that $p(x, y)$ is guaranteed to be a valid probability mass function; i.e., $p(x, y) \geq 0$, $\forall (x, y)$, and $\sum_{(x, y)} p(x, y) = 1$.

The density estimation of (2) can be considered to be the convolution of a 2D filter of impulse response $\phi(x, y)$ with a field of Kronecker impulses situated at $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$. If the region of support $\mathcal{R}$ of window $\phi(x, y)$ is causal, then this convolution can be calculated via a single raster scan through the coefficients. In this case, the causal region of support is $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_2$, where $\mathcal{R}_1 = \{(x, y) : x = 0, y > 0\}$ and $\mathcal{R}_2 = \{(x, y) : x > 0, y \in \mathcal{Z}\}$. By not including $(x, y) = (0, 0)$ in $\mathcal{R}$, both an encoder and its corresponding decoder can make the same estimate of $p(x, y)$ by single raster scan since (2) depends on only values encountered strictly before the current coefficient in the raster scan. In order for (4) to hold for the Laplacian $\phi(x, y)$ of (3) and the causal $\mathcal{R}$, it can be derived that $\beta = \frac{(1-\alpha)^2}{2\alpha}$.

3. THE TARP CODER

In [4], it is proposed to use the density estimate of (2) to code significance of wavelet coefficients in image coding. Specifically, the significance state of a set of coefficients for a given threshold is coded via a raster scan through the coefficients. For coding efficiency, an entropy coder codes $v(x, y)$ for each coefficient, using the probability that $v(x, y) = 1$ for the current coefficient as determined by the density-estimation procedure. The coder of [4] implements the 2D convolution of (2) as a sequence of 1D filtering operations coined *tarp filtering*.¹ This 1D-filtering approach is more efficient than a direct implementation of (2) in that only a limited number of probability estimates need be buffered, and that, because probability estimates are propagated from coefficient to coefficient, fewer arithmetic operations are performed.

In essence, the tarp coder of [4] uses three 1D filters to implement the density estimate of (2)—one filter processes each row from left to right, another filter processes each row from right to

left, and a third filter processes each column from top to bottom. Using this three-filter procedure to estimate significance of wavelet coefficients, the tarp coder is built upon the usual bitplane-coding paradigm—significance and refinement passes are applied successively, and the significance threshold decreases after each pass. The significance pass uses the tarp filter to drive a nonadaptive binary arithmetic coder to code $v(x, y)$ in each subband, while coefficient-sign and refinement information is coded using a nonadaptive binary arithmetic coder on a uniform distribution.

4. THE SHAPE-ADAPTIVE TARP CODER

We modify the tarp coder of [4] to code coefficients arising from an arbitrarily shaped SA-DWT. To do so, our shape-adaptive tarp coder must handle arbitrarily occurring transparent regions in the image. The straightforward approach would be to consider transparent regions to be insignificant as has been done for previous coders. However, keeping in line with the philosophy that we should minimize the number of assumptions we impose on information we do not know, we instead adopt the approach of propagating the current probability of significance, unchanged, across transparent regions.

Our shape-adaptive tarp algorithm is given below. Subband i has size $N_i \times N_i$ and contains coefficients $c_i(x, y)$ with corresponding significance state $v_i(x, y)$ according to the current threshold t . $m_i(x, y)$ is the transparency mask for subband i , and $\text{ac_output}(b, p)$ is a nonadaptive binary arithmetic coder that outputs bit b using probability p that $b = 1$.

```

/* Significance pass */
for each subband i
  for x = 0, ..., Ni - 1
    p2(x) = 0
  endfor
  for y = 0, ..., Ni - 1
    p1 = 0
    for x = 0, ..., Ni - 1
      if mi(x, y) is not transparent
        p = α(p1 + p2(x))
        if vi(x, y) = 0
          if |ci(x, y)| ≥ t
            ac_output(1, p)
            ac_output(sign[ci(x, y)], 0.5)
            ci(x, y) = |ci(x, y)| - t
            vi(x, y) = 1
          else
            ac_output(0, p)
          endif
        endif
        p1 = αp1 + βvi(x, y)
      endif
      p2(x) = p1 + αp2(x)
    endfor
    p3 = 0
    for x = 0, ..., Ni - 1
      p2(x) = p2(x) + αp3
      if mi(x, y) is not transparent
        p3 = αp3 + βvi(x, y)
      endif
    endfor
  endfor
endfor

```

¹The name *tarp filtering* comes from the shape of the Laplacian window of (3) which resembles a tarp draped over a pole.

```

/* Refinement pass */
for each subband  $i$ 
  for  $x = 0, \dots, N_i - 1$ 
    for  $y = 0, \dots, N_i - 1$ 
      if  $m_i(x, y)$  is not transparent
        if  $v_i(x, y) = 1$  (but did not become 1 in the immediately
          preceding significance pass)
          if  $c_i(x, y) \geq t$ 
            ac_output(1, 0.5)
             $c_i(x, y) = c_i(x, y) - t$ 
          else
            ac_output(0, 0.5)
          endif
        endif
      endif
    endfor
  endfor
endfor

```

After each iteration of significance pass and refinement pass, the threshold is decreased, i.e., $t \leftarrow t/2$, and the procedure is repeated until the desired bit rate is achieved. In this implementation, the recursive update of variable p_1 forms the left-to-right row filter, the updating of p_3 corresponds to the right-to-left row filter, and the updating of p_2 implements the top-to-bottom filter carried out on each column. As can be seen, to handle transparent regions, both the p_1 and p_3 values cease updating as the transparent region is crossed. Although it is possible to discontinue the updating of p_2 in the transparent region, our empirical results indicate that it is better to allow p_2 to continue to track p_1 and p_3 throughout the transparent region. Note that, in any event, no assumption on the significant state $v_i(x, y)$ is employed in the algorithm during traversal of a transparent region.

5. EXPERIMENTAL RESULTS

We compare the performance of our shape-adaptive tarp coder to that of shape-adaptive SPIHT [2] and shape-adaptive EBCOT [3] for a variety of still-image objects. We use the QccPack [6] implementations of tarp and SPIHT; these implementations support shape-adaptive coding as described above for tarp and in [2] for SPIHT. We use the implementation of shape-adaptive EBCOT described in [3]. All three techniques employ a 4-scale transform using the CDF 9/7 biorthogonal filter. For the tarp coder, we use $\alpha = 0.6$ for all results—although an “optimal” α could be selected experimentally for each image, we use the value $\alpha = 0.6$ as suggested in [4] as being a universally good choice.

As test data, we use a variety of still-image objects which were extracted from single CIF frames of video sequences used in the MPEG-4 development effort. These frames, along with the object masks, are depicted in Fig. 1; in each case, the object masks were generated via manual segmentation. We measure the “complexity” of the object masks with a boundary-to-area ratio (BAR) defined as the total number of mask pixels that lie on the boundary of the object expressed as a proportion of the total number of opaque pixels in the object. BAR values for the masks of Fig. 1 are tabulated in Table 1.

Distortion as calculated over only opaque pixels for a fixed rate is also tabulated in Table 1. We see that our shape-adaptive tarp coder consistently outperforms the shape-adaptive SPIHT coder. In general, we see that the larger the BAR value for the mask, the larger is the performance margin in favor of shape-adaptive

tarp, ranging from a 1-dB gain for objects with complex masks (i.e., hall-monitor with BAR = 14.2%) to a 0.5-dB gain for simpler masks (i.e., erik with BAR = 2.0%). On the other hand, in comparison to shape-adaptive EBCOT, the performance is more mixed, with both techniques producing more or less equivalent rate-distortion performance for most images. The only anomaly is the coastguard image—for this image, the shape-adaptive EBCOT technique significantly outperforms our tarp coder. Figs. 2 and 3 give the rate-distortion performance for the hall-monitor (complex mask) and penguin datasets (simple mask), respectively, over a range of rates.

6. CONCLUSIONS

When the full frames of the images in Fig. 1 are coded at 0.5 bpp, the tarp coder averages some 0.3 dB less PSNR than SPIHT and some 1.1 dB less PSNR than EBCOT. However, when converted into shape-adaptive coders, both SPIHT and EBCOT impose the assumption that transparent regions surrounding image objects are insignificant coefficients. Because these coders are forced to directly or indirectly incorporate this artificial data while coding opaque coefficients near object boundaries, our tarp coder, which avoids imposing this assumption on the significance state of unknown transparent regions, becomes much more competitive when shape-adaptive coding is considered. Specifically, our shape-adaptive tarp coder provides rate-distortion performance significantly superior to that of shape-adaptive SPIHT and roughly equivalent to that of shape-adaptive EBCOT. However, it must be noted that the tarp algorithm is significantly easier to implement than either SPIHT or EBCOT. Additionally, the tarp algorithm is substantially less computationally complex than EBCOT—our informal observations indicate that our shape-adaptive tarp coder runs some three times faster than the shape-adaptive EBCOT implementation from [3].

7. ACKNOWLEDGMENT

Thanks to Z. Xiong for providing the implementation of shape-adaptive EBCOT from [3].

8. REFERENCES

- [1] S. Li and W. Li, “Shape-adaptive discrete wavelet transforms for arbitrary shaped visual object coding,” *IEEE Trans. Cir. Sys. Video Tech.*, vol. 10, no. 5, pp. 725–743, Aug. 2000.
- [2] G. Minami, Z. Xiong, A. Wang, and S. Mehrota, “3-D wavelet coding of video with arbitrary regions of support,” *IEEE Trans. Cir. Sys. Video Tech.*, vol. 11, no. 9, pp. 1063–1068, Sep. 2001.
- [3] Z. Liu, J. Hua, Z. Xiong, and K. Castleman, “Lossy-to-lossless ROI coding of chromosome images using modified SPIHT and EBCOT,” in *Proc. IEEE Inter. Symp. Biomedical Imag.*, Washington, DC, Jul. 2002, pp. 317–320.
- [4] P. Simard, D. Steinkraus, and H. Malvar, “On-line adaptation in image coding with a 2-D tarp filter,” in *Proc. IEEE Data Comp. Conf.*, J. A. Storer and M. Cohn, Eds., Snowbird, UT, Apr. 2002, pp. 23–32.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, Inc., New York, 2nd edition, 2001.
- [6] J. E. Fowler, “QccPack: An open-source software library for quantization, compression, and coding,” in *App. Digital Image Proc. XXIII*, A. G. Tescher, Ed., San Diego, CA, Aug. 2000, Proc. SPIE 4115, pp. 294–301.

Figure 1: Original images and corresponding object masks. From top to bottom: coastguard (frame 250), road-surveillance1 (frame 211), hall-monitor (frame 128), penguin (frame 211), paris (frame 111), erik (frame 34)

Image	BAR	PSNR (dB)		
		tarp	SPIHT	EBCOT
coastguard	14.6%	19.9	19.3	20.8
road-surveillance1	14.5%	23.7	22.8	23.7
hall-monitor	14.2%	22.9	21.9	22.7
penguin	5.7%	31.1	30.5	31.4
paris	3.3%	28.3	27.9	27.9
erik	2.0%	28.0	27.5	27.9

Table 1: Mask complexities for the datasets and distortion performance for the shape-adaptive coders. Nominal rate is 0.5 bpp (tarp and SPIHT meet rate exactly; actual rate for EBCOT is highest possible rate not exceeding 0.5 bpp).

Figure 2: Rate-distortion performance for hall-monitor.

Figure 3: Rate-distortion performance for penguin.