A NEURAL NETWORK FOR CONTEXT-BASED ARITHMETIC CODING IN LOSSLESS IMAGE COMPRESSION

G.A. Triantafyllidis and M.G. Strintzis

Information Processing Laboratory, Aristotle University of Thessaloniki, Thessaloniki 540 06, Greece
Tel. : +3031.996351, Fax : +3031.996342

ABSTRACT
Significant progress has recently been made in lossless image compression using discrete wavelet transforms. The overall performance of these schemes may be further improved by properly designing efficient entropy coders. In this paper a new technique is introduced for the implementation of context-based adaptive arithmetic entropy coding. This technique is based on the prediction of the value of the current transform coefficient, using a neural network, in order to achieve appropriate context selection for arithmetic coding. Experimental results illustrate and evaluate the performance of the proposed technique.

1. INTRODUCTION
Two entropy coding methods are well-known and widely used: the Huffman and the arithmetic. The first method is preferable only when there is a lack of hardware resources and coding/decoding speed is a prime objective [1]. Arithmetic is somewhat slower than Huffman, but it is much more versatile and effective. In most cases, the adaptive variant of arithmetic coding is used [2],[3], in order to take advantage from high order dependencies with the use of conditioning contexts.

The arithmetic data compression technique encodes data by creating code string which represents a fractional value on the number line between 0 and 1. On each recursion of the algorithm only one symbol is encoded. The algorithm successively partitions an interval of the number line between 0 and 1, and retains one of the partitions as a new interval. Thus, the algorithm successively deals with smaller intervals, and the code string lies in each of the nested intervals.

The performance of arithmetic coders depends mainly on the estimation of the probability model which the coder will use. If the probability model accurately reflects the statistical properties of the input, arithmetic coding will approach the entropy of the source. Different probability models will give different compression performance for the same data and thus the probabilities that an adaptive model assigns may change as each symbol is transmitted, based on the symbol frequencies seen so far in the message. A drawback of arithmetic coding of images using the above adaptive model is that it does not take into account the high amount of correlation between adjacent pixels. That is, each pixel is encoded using a probabilistic model adapted to all pixel values seen so far on the image. In this work, to alleviate this disadvantage a method similar to the one in [4] is adopted, with which, for every new coefficient to be encoded, the model is updated more than once, making the probabilistic model more adaptive to recent pixels, and thus more effective.

Every transform coefficient is put into one of several classes (buckets) depending on the weighted values of a set of previously entropy coded coefficients. To each context type corresponds a different probability model and thus each subband coefficient is compressed with an entropy coder following the appropriate model. The key issue is then how to find an efficient context based classification.

The paper is organized as follows: In Section 2 a simple neural network is developed to determine the prediction of the current coefficient. Section 3 presents experimental results and conclusions are drawn in Section 4.
2. PREDICTION AND CONTEXT SELECTION

In our work, the Magnitude-Set Variable-Length-Integer representation (proposed in [5] and shown in Table 1) is employed to represent the transform coefficients. According to this, every coefficient is classified into one of a set of ranges called magnitude sets $M$, followed by the sign bit and the magnitude difference bits. For example, the numbers 15 and -16 are transmitted with the number triads $(7, +, 3)$ and $(8, - , 0)$ respectively. The magnitude set $M$ of the current

<table>
<thead>
<tr>
<th>Magnitude Set</th>
<th>Amplitude Intervals</th>
<th>Sign Bit</th>
<th>Magnitude Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[0]</td>
<td>no</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>[-1][1]</td>
<td>yes</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>[-2][2]</td>
<td>yes</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>[-3][3]</td>
<td>yes</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>[-5,-4],[4,5]</td>
<td>yes</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>[-7,-6],[6,7]</td>
<td>yes</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>[-11,-8],[8,11]</td>
<td>yes</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>[-15,-12],[12,15]</td>
<td>yes</td>
<td>2</td>
</tr>
<tr>
<td>8</td>
<td>[-23,-16],[16,23]</td>
<td>yes</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 1. Definition of Magnitude-Set Variable-Length-Integer ($MS = VLI$) representation

pixel is estimated using the weighted values of coefficients that have already been entropy encoded in the current band, in the sister band(s) and in the parent band, in the pyramid structure, i.e., the predictor has the form:

$$M = \sum_{i=0}^{N} a_i M_i \quad (1)$$

where the $M_i$ indicates a previously encoded Magnitude Set, $M$ is the prediction of the current Magnitude Set and the weights $a_i$, $1 \leq i \leq N$ are determined so that $M$ are estimates of $M$.

Experimental results have proved that the magnitude sets of the coefficients shown in Figure 1, which differ in shape for every subband LH, HL and HH, suffice for an accurate prediction of the magnitude set $M$ of the current pixel. Therefore, equation 1 is expressed for each subband as follows [11]:

$LH$ band:  
$$M = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} + a_5 M_{p1} + a_6 M_{p2}$$

$HL$ band:  
$$M = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} + a_5 M_{p1} + a_6 M_{p2} + a_7 M_{si s}$$

$HH$ band:  
$$M = a_1 M_w + a_2 M_{nw} + a_3 M_n + a_4 M_{ne} + a_5 M_{p1} + a_6 M_{p2} + a_7 M_{si s1} + a_8 M_{si s2} \quad (2)$$

Subscripts $w, nw, n, ne$ are directional short notations for west, north-west, north and north-east respectively, $p_k$ ($k = 1, 2$) indicates the $k^{th}$ parent pixel and $s_i$'s indicate the corresponding pixels or pixel in the sister bands.

![Fig. 1. Pixels employed for the prediction of the magnitude set of current coefficient for each subband](image)

2.1. Weight Optimization via Neural Network

A new method is proposed for the calculation of the optimal weights used for the proper context selection, based on a neural network. A neural network is implemented in order to find the fittest possible weights to make an estimation for the current coefficient. It is quite obvious that the use of a proper neural network can lead to better estimation of the $MS$ of the current coefficient compared to that obtained from the simple least squares method which was used in [11]. In other words, it may be achieved a more efficient context-based classification, resulting in improvement
of the compression performance of the arithmetic entropy encoder.

In our experiments, a simple Back-Propagation Neural Network, which is drawn in Figure 2 and fully described in [8], is put into practice in order to perform the weight optimization. This Neural Network is quite simple and involves only the weights $W_i, V_h, \Theta_j, \Gamma_i$, where $i = 1, ..., p$, $\Gamma_j, \Theta_h = 1, ..., q$, $h = 1, ..., n$.

For each image that needs to be encoded, this neural network has to be trained in order to conclude to the fittest possible weights and bias factors. However, if we want, we can go one step further: to train the neural network with a whole set of typical images. In that case, it is clear that compression performance is going to be decreased. Using the optimal weights calculated by the neural network, we can classify the $MS$ of the current pixel to the appropriate bucket for the arithmetic coding.

![Fig. 2. A simple Back-Propagation Neural Network](image)

### 3. EXPERIMENTAL RESULTS

The above context-based arithmetic entropy coding technique was compared to the method used in the widely regarded as state-of-the-art algorithm of S+P [5]. Our experiments may be summarized as follows:

<table>
<thead>
<tr>
<th>image</th>
<th>S+P bytes</th>
<th>method I bytes</th>
<th>bpp S+P</th>
<th>bpp method I</th>
</tr>
</thead>
<tbody>
<tr>
<td>lena</td>
<td>136702</td>
<td>135973</td>
<td>4.17</td>
<td>4.15</td>
</tr>
<tr>
<td>peppers</td>
<td>150182</td>
<td>149159</td>
<td>4.58</td>
<td>4.55</td>
</tr>
<tr>
<td>crowd</td>
<td>131010</td>
<td>130329</td>
<td>4.00</td>
<td>3.98</td>
</tr>
<tr>
<td>boat</td>
<td>141272</td>
<td>139817</td>
<td>4.31</td>
<td>4.26</td>
</tr>
<tr>
<td>airplane</td>
<td>128238</td>
<td>126967</td>
<td>3.91</td>
<td>3.87</td>
</tr>
<tr>
<td>bridges</td>
<td>182882</td>
<td>182001</td>
<td>5.58</td>
<td>5.55</td>
</tr>
<tr>
<td>harbour</td>
<td>154896</td>
<td>152993</td>
<td>4.73</td>
<td>4.67</td>
</tr>
<tr>
<td>barbara</td>
<td>149254</td>
<td>147789</td>
<td>4.55</td>
<td>4.51</td>
</tr>
</tbody>
</table>

Table 2. Number of bytes and bits per pixel needed for entropy coding with optimal weights calculated via a neural network compared to S+P entropy coding.

4. CONCLUSIONS

A neural network was developed for the implementation of an efficient context-based arithmetic entropy coding. The method employs the neural network to estimate the magnitude set of the current coefficient, based on a selected set of magnitude sets of pixels which have been previously coded. Experiments show that the use of the proposed algorithm leads to better results, and consistently outperforms the entropy coder proposed by the state-of-the-art method S+P in [5].
5. REFERENCES


