Efficient lossless compression of colour image data with diverse characteristics

Tilo Strutz, Thomas Schmiedel and Thomas Schatton
Deutsche Telekom AG
Hochschule für Telekommunikation Leipzig, Institute of Communications Engineering
Gustav-Freytag-Str. 43–45, 04277 Leipzig, GERMANY

Abstract—Lossless compression of data is the attempt to model the data in such a manner that the resulting file contains the pure information without any redundancy. This paper presents a universal compression system which is able to compress colour images efficiently with a broad range of characteristics from natural images to synthetic images with sparse histograms. The proposed compression scheme uses a novel adaptive parameter selection and a pre-coding stage, which utilizes not only local correlation of the signal values, but also exploits long-distance dependencies. The new approach was tested against several state-of-the-art compressors for a set of eighteen images demonstrating that the average compression efficiency of the proposed scheme is superior.

I. INTRODUCTION

The reversible compression of image requires invertible processing steps. In general, this characteristic is achieved by using processing steps which map integer input samples to integer output values, particularly the decorrelation step. Successful compression schemes either predict the pixel values based on other pixels which have already been processed or apply integer transforms such as the integer wavelet transform (IWT) [1]. Typically, the lossless compression performance based on prediction is higher than for wavelet-based systems (see for example [2]).

In prediction-based systems, the prediction errors are typically entropy coded. The remaining statistical dependencies are utilised by context-adaptive switching between different coding tables (e.g. [3], [4]). In combination with arithmetic coding, the modelling of the error distribution (e.g. [5]–[11]) and even the blending of the distributions of different predictors used are also promising (e.g. [12]).

In the past, research on lossless image coding was mostly focussed on the compression of natural grey-scale images such as photographs. There are, however, many computer-generated images which have a more synthetic characteristic, which cannot be compressed efficiently with the same compression techniques. Typically, the portable network graphics (PNG) format is the container of choice for storing synthetic images. Unfortunately, PNG does not compress natural images very well. Moreover, images are not only purely natural or synthetic, such as simple diagrams or charts; instead, the full spectrum also contains mixtures of computer-generated image content and photographs, as well as rendered images, which are computer-generated but look like natural scenes.

II. METHOD

A. General overview

Figure 1 shows the stages of the encoding process. It
starts with an adaptive colour-space conversion. An analysis tool checks which of the four colour spaces – RGB (i.e. no conversion), YUV, YCgCo-R or HP2 – yields the best decorrelation of the data by means of lowest entropy. YUV corresponds to the reversible colour transform of JPEG2000 [14]

\[
U_r = B - G, \quad V_r = R - G, \quad Y_r = G + \left\lfloor \frac{U_r + V_r}{4} \right\rfloor, \quad (1)
\]

while YCgCo-R is a proposal for the fidelity range extension of the video coding standard H.264 [15]

\[
C_0 = R - B, \quad t = B + \lfloor C_0/2 \rfloor, \\
C_g = G - t, \quad Y = t + \lfloor C_g/2 \rfloor. \quad (2)
\]

HP2 is defined in the CharLS project [16] as

\[
Y = G, \quad U = B - \lfloor (R + G)/2 \rfloor, \quad V = R - G. \quad (3)
\]

All methods convert the image data without loss of information. The decision pro colour space conversion can be withdrawn by the automated parameter selection under certain circumstances (see below).

The prediction is performed separately for each colour component. The image components are segmented in blocks of 24 × 24 pixels and the best predictor out of five predictors is selected, including

- Median Edge Detector (MED) [3],
- Gradient-Adjusted Predictor (GAP) [6],
- Paeth Predictor [17],
- simple linear prediction,
- context-based adaptive linear prediction (CoBALP) [4].

The choice of the prediction method must be transmitted along the coded data as additional information.

The processing in the pre-coding stage is one-dimensional and requires a certain interleaving of the colour components. In principle, it would be possible to process the components independently from one another (principle, it would be possible to process the components in dendence, and requires a certain interleaving of the colour components. In this case, the components are interleaved sample by sample.

The pre-coding stage is a sequence of run-length coding (RLC), block-sorting, ranking and a second run-length coding (Figure 1). The RLC 1 reduces the number of symbols to be transmitted along the coded data as additional information. The final processing step encodes the output symbols of the last RLC. Two Huffman codes are adaptively constructed, one for the symbols (including the token) and one for the run lengths. The code information is transmitted together with the coded data.

All processing steps are fully reversible and the corresponding decoding algorithm reconstructs the image data in reverse order.

Figure 2 shows an example how the distribution becomes narrower during the processing steps. To enable comparison, the prediction-error signal and the RLC1 output were folded into the non-negative range in an interlaced manner. In this example, the RLC 1 mainly compacts runs of zero, as can be concluded from the ‘Prediction’ and ‘RLC 1’ curves. After block sorting and ranking, the distribution is narrower and the number of zeros is increased. The second run-length coding stage reduces the total number of symbols to be transmitted by compacting the newly created runs of zeros.
B. Automated parameter selection

The compression of data is optimal when the data model is understood as best as possible and this model can be addressed with an appropriate compression method. The proposed compression system shows some degree of freedom in selecting suitable processing parameters, which includes the colour space, the interleaving of colour components (both already discussed above), the prediction method and the pre-coding technique in terms of using either move-to-front (MTF) coding or iterative frequency count (IFC).

The parameter selection is controlled by only two properties of the image: the number of colours (and its relation to the size of the image) and the ‘syntheticness’. The latter is explained below.

1) Categorisation of images: There are at least three general types of images with respect to data compression: images which should be processed using a colour map, images which should be decorrelated using colour-space transform and prediction¹ and images which should be processed without a spatial decorrelation step. In principle, the categorisation could be done based on the number of pixels \( N \) and the number of unique colours \( C \) in the image (Figure 3). Please note that the diagram is not true to scale.

If, for example, the quotient \( C/N \) is below a threshold and the number of colours is also below a certain number, then spatial decorrelation techniques (e.g. prediction) most likely will have adverse effects on compression performance.

2) Syntheticness of images: Categorisation based on number of colours \( C \) and image size \( N \) is not entirely sufficient for optimal parameter selection. In addition, a measure of syntheticness is proposed, which is based on the characteristic of the colour component’s histograms \( \{ h_r \}, \{ h_g \} \) and \( \{ h_b \} \), whereby \( r, g, \) and \( b \) stand for the intensity values of red, green and blue. The measure sums up the absolute differences between adjacent frequency counts \( h[x] \) with \( 0 \leq x \leq 255 \), for components with 8 bits per pixel

\[
S_x = \frac{1}{N} \sum_{x=0}^{254} |h[x+1] - h[x]| + |h[0] - h[255]|, \tag{4}
\]

where \( N \) is the image size. Value \( S = (S_r + S_g + S_b)/3 \) is in the range \([0, 2]\). The larger \( S \), the more distinct peaks and gaps the histogram contains and the more likely the image is of synthetic nature. Some images have a distinct background colour. The corresponding intensity value \( x \) corresponds to a significant peak in the histogram and would bias \( S \) towards large values. The influence of such a background value is suppressed by excluding the modal value \( x_{\text{mod}} = \arg\max_x(h[x]) \) from the summation in equation (4) and correcting the number of pixels accordingly.

3) Derived processing steps: Using the characterisation of images as discussed above, the following compression parameters are used in the current implementation. If the number of unique colours \( C \) is lower than the fixed number of 100000, \( C/N < 0.032 \) and synthethics is larger than 0.1, then the image is processed using a colour map (which must be included in the bitstream), without prediction and using plain MTF coding after the BWT. If one of the three conditions is not met, the best colour space is determined as described in Subsection II-A. Further, the parameters are tested against the following thresholds: \( C < 90000, C/N < 0.055 \) and \( S > 0.1 \). If the conditions are met, the prediction is bypassed and the interleaving mode YuvYuv and MTF coding are used. In all other cases, the block-based prediction is applied. If colour space conversion is enabled, YYuvYuv interleaving and IFC are activated. Otherwise (i.e. RGB is preferred), the synthethics determines the other coding parameters. If \( S > 0.9 \), then YuvYuv interleaving and MTF are used; otherwise YYuvYuv interleaving and IFC are activated.

4) Coding of the colour map: Since prediction is disabled when indexed colours are used, the map items can be sorted in a manner that allows efficient compression of the colour map. The order of the colours has only marginal influence on the performance of the pre-coding steps described above.

The new coding scheme computes a single colour value from the \((R,G,B)\) triple

\[ c_{\text{RGB}} = G \cdot 2^{16} + R \cdot 2^8 + B . \tag{5} \]

The values \( c_{\text{RGB}} \) are sorted in ascending order, with the effect that the list of G values starts with a small number followed by identical or slightly increased values. The differences between adjacent G values, which are always non-negative, are fed into an adaptive Rice encoder. The coding of the list of R values follows the same principle, as long as the G values do not change. As soon as a new G value appears, the encoder switches to another context and the R value is processed in a non-differential manner. The same holds true for the B values, with the distinction that both R and G must be constant as a prerequisite for the differential method. The colour map of the image ‘meditate_fel’ (see next section), for example, which contains 32096 unique colours, requires only 29528

Fig. 3. Categorisation of images in terms of compression parameters dependent on image size \( N \) (number of pixels) and number of unique colours \( C \)

¹Alternatively, the spatial decorrelation could be performed using a signal transform such as discrete wavelet or cosine transform.
bytes (4044 (G) + 4752 (R) + 19732 (B)), making the indexed compression the most efficient method for this image. To the author's knowledge, compression of images with colour maps having more than 256 entries was not considered yet.

III. COMPRESSION RESULTS

The performance of the new compression algorithm was tested based on 18 images [21] with different characteristics, mostly taken from the internet: ‘p08’, ‘woman’ and ‘bike’ from a JPEG test set provided by Thomas Richter. Eight of the images are photographs or scanned samples and three are ray-traced images, while the others are charts, screen dumps or other computer-generated images.

Table I contains the compression results by means of bits per pixel. Column ‘S’ shows the value resulting from the analysis of the image histograms, eq.(4), and C is the number of unique colours. The proposed algorithm is called TSIP in the style of ‘ZIP’ for general lossless compression. ‘Auto’ means that all parameters of the algorithm are set automatically as described above; ‘opt.’ corresponds to tuned parameters. These are mainly different colour spaces and different block sizes for block-based prediction, but also other predictors or different interleaving modes. In particular, if an image shows a mixture of synthetic and natural content, the decision for or against a particular strategy might lead to suboptimal compression results.

The results are compared to ‘PNGOUT’ (ırfanView, plugin PNGOUT, default options), LOCO-I [22], and ‘7zip’ (version 9.20 with compression mode ‘Ultra’). LOCO-I is the core algorithm of JPEG-LS [3]. Its results are obtained using the same colour transform, as automatically chosen by TSIP (Table I, column ‘colour’). For each image, the best result is printed in bold numbers and the second best result is underlined.

With regard to the tested images, the new TSIP compression scheme is almost always among the top two algorithms and requires the smallest number of bits per pixel on average. The image ‘png-8passes’ is the only example in which the proposed compression scheme performs distinctly worse than the best one (7zip). This fact is somewhat surprising, since TSIP performs comparably to or better than 7zip for the majority of tested images.

The proposed compression scheme is able to switch automatically to a mode with indexed colours. It should be said that there are dedicated algorithms for images with low number of unique colours, such as the piecewise-constant image model [23] and context-tree modelling [24]. These methods could not be included in the test due to the lack of an appropriate software implementation.

The compression time on a standard desktop computer varies for TSIP from few seconds up to several minutes, when BWT faces a high sorting depth due to many identical symbols. The source code, however, is not optimised for speed yet.

IV. SUMMARY AND DISCUSSION

We have proposed a novel coding scheme for the lossless compression of arbitrary colour images. In contrast to other compression systems focussing on natural images like photographs, the coding stage of the new scheme does not rely on the local two-dimensional correlation of prediction errors, but it is able to exploit long-distance correlations. This makes the new approach applicable to images with synthetic content as well, while high compression ratios can be obtained regardless of the image type.

The compression scheme described here distinguishes itself through image analysis-driven selection of parameter settings (colour space, prediction method, interleaving mode and pre-coding technique). The different interleaving techniques enable the exploitation of remaining correlations between the colour components. When using a YUV-like colour space, the combined treatment of the U and V components (i.e. using YY(uvuv mode) benefits the compression for most images. Without any colour transform, full interleaving (YuvYuv) typically yields the best results.

The investigations have shown that compression of images with colour maps containing far more than 256 unique colours can be more efficient than processing with three separate components. Using the proposed coding method for colour maps, even the LOCO-I algorithm compresses the synthetic image ‘europa_karte_de’ just as efficiently as 7zip.

Use of the Burrows-Wheeler transform (BWT) inhibits the exploitation of two-dimensional correlation, which means the performance of the new scheme cannot outperform specialised systems such as LOCO-I for images without any long-distance correlation. Based on a proper image analysis, however, the LOCO-I algorithm could be selected adaptively.

The computation time is a drawback of the BWT. Even if the source code were optimised for speed, sorting would last longer than simple processing in LOCO-I. Future investigations will consider variants of block sorting with limited sorting depth [25].

Column ‘TSIP (opt.)’ in Table I shows that there is room for improvement in automated parameter selection. The block size for block-based predictor selection should be dependent at least on the image size, for example, but content-adaptive strategies using quadtrees might also be a promising option [26]. In addition, the re-scaling of symbol counts in the IFC step could be adapted to the image properties. In general, a more sophisticated analysis of the image or immediate signals is needed. The segmentation of images having content with varying properties is another problem which must be dealt with in future.

Finally, it should be mentioned that the implemented entropy-coding stage could be improved by replacing the Huffman coding with an adaptive arithmetic-coding stage. This not only drops the amount of compressed data but also saves some bits required for the transmission of the Huffman-code tables.
<table>
<thead>
<tr>
<th>image</th>
<th>size</th>
<th>S</th>
<th>C</th>
<th>PNGOUT</th>
<th>7zip</th>
<th>colour</th>
<th>LOCO-I</th>
<th>TSIP (auto)</th>
<th>TSIP (opt.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>boots</td>
<td>1018 × 854</td>
<td>0.03</td>
<td>4895</td>
<td>9.040</td>
<td>7.685</td>
<td>HP2</td>
<td>7.332</td>
<td>6.580</td>
<td>6.546</td>
</tr>
<tr>
<td>p08</td>
<td>2592 × 1944</td>
<td>0.04</td>
<td>32410</td>
<td>12.442</td>
<td>14.493</td>
<td>HP2</td>
<td>7.884</td>
<td>8.097</td>
<td>7.970</td>
</tr>
<tr>
<td>blockMap</td>
<td>1430 × 2012</td>
<td>0.05</td>
<td>32093</td>
<td>14.590</td>
<td>16.937</td>
<td>YUVr</td>
<td>12.508</td>
<td>12.592</td>
<td>12.413</td>
</tr>
<tr>
<td>barbara2</td>
<td>720 × 576</td>
<td>0.05</td>
<td>100013</td>
<td>15.915</td>
<td>17.804</td>
<td>YUVr</td>
<td>10.197</td>
<td>9.650</td>
<td>9.584</td>
</tr>
<tr>
<td>party_4s</td>
<td>1600 × 1200</td>
<td>0.07</td>
<td>447865</td>
<td>12.685</td>
<td>14.607</td>
<td>YCoCg-R</td>
<td>10.429</td>
<td>10.984</td>
<td>10.822</td>
</tr>
<tr>
<td>kodim23</td>
<td>768 × 512</td>
<td>0.11</td>
<td>72079</td>
<td>11.252</td>
<td>12.821</td>
<td>YUVr</td>
<td>8.580</td>
<td>8.859</td>
<td>8.817</td>
</tr>
<tr>
<td>png-8passes</td>
<td>1024 × 768</td>
<td>0.12</td>
<td>43032</td>
<td>5.922</td>
<td>2.684</td>
<td>Indexed</td>
<td>12.162</td>
<td>3.742</td>
<td>3.742</td>
</tr>
<tr>
<td>woman.rgb</td>
<td>2048 × 2560</td>
<td>0.19</td>
<td>99642</td>
<td>14.851</td>
<td>13.223</td>
<td>Indexed</td>
<td>11.346</td>
<td>10.946</td>
<td>10.825</td>
</tr>
<tr>
<td>bike</td>
<td>2048 × 2560</td>
<td>0.24</td>
<td>271926</td>
<td>14.643</td>
<td>12.333</td>
<td>YCoCg-R</td>
<td>11.509</td>
<td>12.108</td>
<td>11.859</td>
</tr>
<tr>
<td>cwheel</td>
<td>800 × 600</td>
<td>0.26</td>
<td>83172</td>
<td>3.713</td>
<td>5.021</td>
<td>RGB</td>
<td>4.004</td>
<td>3.687</td>
<td>3.513</td>
</tr>
<tr>
<td>woodgrove</td>
<td>1149 × 852</td>
<td>0.28</td>
<td>56177</td>
<td>3.376</td>
<td>3.309</td>
<td>YCoCg-R</td>
<td>3.873</td>
<td>3.378</td>
<td>3.074</td>
</tr>
<tr>
<td>artificial18</td>
<td>800 × 533</td>
<td>0.31</td>
<td>39584</td>
<td>5.106</td>
<td>4.512</td>
<td>YCoCg-R</td>
<td>4.623</td>
<td>4.722</td>
<td>4.305</td>
</tr>
<tr>
<td>bee_teacher</td>
<td>4825 × 972</td>
<td>0.42</td>
<td>266424</td>
<td>4.786</td>
<td>4.995</td>
<td>YUVr</td>
<td>4.302</td>
<td>4.294</td>
<td>4.163</td>
</tr>
<tr>
<td>dev</td>
<td>800 × 619</td>
<td>0.67</td>
<td>17565</td>
<td>1.872</td>
<td>1.424</td>
<td>YUVr</td>
<td>1.803</td>
<td>1.694</td>
<td>1.604</td>
</tr>
<tr>
<td>nur_pulincier</td>
<td>1112 × 720</td>
<td>1.11</td>
<td>56477</td>
<td>3.062</td>
<td>2.739</td>
<td>YUVr</td>
<td>2.844</td>
<td>2.497</td>
<td>2.444</td>
</tr>
<tr>
<td>meditate_fel</td>
<td>3000 × 2248</td>
<td>1.53</td>
<td>32096</td>
<td>1.963</td>
<td>1.194</td>
<td>Indexed</td>
<td>1.307</td>
<td>0.954</td>
<td>0.940</td>
</tr>
<tr>
<td>europa_karte_de</td>
<td>1200 × 1000</td>
<td>1.69</td>
<td>52</td>
<td>0.600</td>
<td>0.663</td>
<td>Indexed</td>
<td>0.676</td>
<td>0.525</td>
<td>0.512</td>
</tr>
<tr>
<td>light_world-1</td>
<td>4096 × 4096</td>
<td>1.99</td>
<td>401</td>
<td>1.057</td>
<td>0.462</td>
<td>Indexed</td>
<td>2.733</td>
<td>0.366</td>
<td>0.366</td>
</tr>
<tr>
<td>averages</td>
<td></td>
<td></td>
<td></td>
<td>7.604</td>
<td>7.606</td>
<td></td>
<td>6.576</td>
<td>5.866</td>
<td>5.750</td>
</tr>
</tbody>
</table>

*This result was obtained using YUVr colour space. The bitrate using indexed colours is higher.

*dito

REFERENCES


[16] apple.com, last visited 15.02.2011


