ADAPTIVE SELECTION OF COLOUR TRANSFORMATIONS FOR REVERSIBLE IMAGE COMPRESSION

Tilo Strutz

Deutsche Telekom, Hochschule für Telekommunikation Leipzig
Institute of Communications Engineering
Gustav-Freytag-Str. 43–45, 04277 Leipzig, Germany

ABSTRACT

This paper presents and investigates a new family of reversible low-complexity colour transformations. It shows that, for a reasonably large set of natural images, there is a colour transform which performs better in the context of lossless image compression than the reversible colour transform defined in the JPEG2000 standard, while having only slightly increased complexity. The optimal selection of a colour space for each single image can distinctly decrease the bitrate of the compressed image. A novel approach is proposed, which automatically selects a suitable colour space with negligible loss of performance compared to the optimal selection.

Index Terms— image compression, reversible colour transform

1. INTRODUCTION

The reversible compression of image requires processing steps which are themselves invertible. This characteristic is achieved, in general, by using processing steps which map integer input samples to integer output values. This also concerns the colour transformation, which aims at decorrelating the colour components Red, Green and Blue (RGB).

A colour transformation converts a triple of non-negative integer values (R, G, B) into another representation, say (Y, U, V), using a 3×3 matrix $\mathbf{A}$. The elements of the matrix should be chosen so that the compression of the image leads to a minimum bitrate

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \mathbf{A} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}. \qquad (1)$$

In [1], it was stated that the implementation with ladder networks (also known as lifting schemes) and triangular matrices with unity diagonal elements enables a mapping from integer RGB values to integer YUV values. Rounding of intermediate values is an essential step here. Most popular reversible colour transformations use this principle as, for example, the reversible colour transformation (YUVr) defined in the JPEG2000 standard [2], and the YCgCo-R colour space as proposed for the fidelity range extension of the video coding standard H.264 [3].

This paper develops a new family of reversible low-complexity colour transformations. To the author's knowledge, this is the first systematic evaluation of a wide variety of colour transformations.

The image-adaptive decorrelation of the colour components has already been addressed in literature, while focussing on different applications [4, 5, 6, 7, 8]. The adaptation is performed either based on the entire image, block-based or pixel-by-pixel. It is obvious that local adaptation of processing steps is most likely superior to global adaptation, especially if the decoder is able to make the decisions without transmitting extra information. Unfortunately, the switching between different colour spaces introduces discontinuities, influencing the subsequent processing. For this reason, this paper concentrates on an approach which selects a single suitable colour space for an entire image. The main purpose is to show the potential of the new family of colour spaces.

The paper is organised as follows: Section 2 reviews two popular colour transforms. Based on this, Section 3 derives a new family of related colour transformations. Section 4 proposes a new scheme for the automatic colour-space selection. Section 5 investigates its application to lossless image compression and Section 6 discusses the results and concludes the paper.

2. REVIEW OF POPULAR REVERSIBLE COLOUR TRANSFORMATIONS

2.1. YUVr colour transform

The reversible colour transform defined in JPEG2000 uses the matrix

$$\mathbf{A}_1 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{pmatrix}. \qquad (2)$$

Together with the permutation matrix

$$\mathbf{P}_1 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}, \qquad (3)$$

a proper factorisation into triangular matrices, enabling the rounding operations necessary for the inner factors, would be

$$\begin{aligned} \mathbf{A}_1 &= \mathbf{P}_1 \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1/4 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \mathbf{P}_1 \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 1/4 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}. \qquad (4) \end{aligned}$$

Figure 1 shows the corresponding signal flow including the inverse transformation. It becomes clear that each single processing step of the forward transformation can be reversed in the back transformation simply by inverting the order of processing and flipping the signs of coefficients. The inverse transformation matrix is

$$(\mathbf{A}_1)^{-1} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ -1/4 & 1 & -1/4 \\ 0 & 0 & 1 \end{pmatrix} \cdot \mathbf{P}_1^{\mathrm{T}} = \begin{pmatrix} 1 & -1/4 & 3/4 \\ 1 & -1/4 & -1/4 \\ 1 & 3/4 & -1/4 \end{pmatrix}.$$

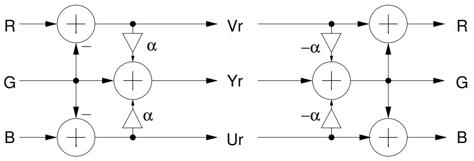

Fig. 1. Processing steps of reversible colour transformation (YUVr) defined in JPEG2000 with $\alpha = 1/4$

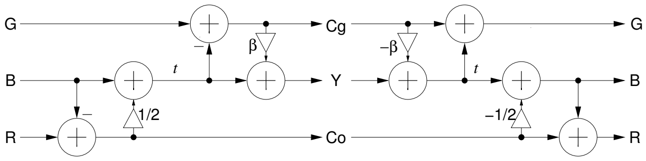

Fig. 2. Processing steps of the YCgCo-R colour transformation and its inverse, $\beta = 1/2$

The integer-to-integer mapping is achieved by rounding the intermediate values of each single step. The complete calculations of the RGB-to-YUVr and YUVr-to-RGB transformations are

$$U_\mathrm{r} = B - G \quad V_\mathrm{r} = R - G \quad Y = G + \lfloor (U_\mathrm{r} + V_\mathrm{r})/4 \rfloor \qquad (5)$$

$$G = Y - \lfloor (U_\mathrm{r} + V_\mathrm{r})/4 \rfloor \quad R = V_\mathrm{r} + G \quad B = U_\mathrm{r} + G\ . \qquad (6)$$

The YUVr colour space shows an excellent decorrelation performance for a broad range of images and obviously has a very low complexity (four additions and one bit shift operation per pixel).

2.2. YCgCo-R colour transform

The RGB-to-YCgCo-R transformation, [3], has a likewise low complexity; the factorisation, however, is slightly different

$$\mathbf{C}_1 = \mathbf{P}_2 \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1/2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1/2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 0.25 & 0.5 & 0.25 \\ -0.5 & 1 & -0.5 \\ 1 & 0 & -1 \end{pmatrix}, \qquad \text{with } \mathbf{P}_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}. \qquad (7)$$

Figure 2 shows the corresponding signal flow for the forward and the inverse transformation. Note that both the RGB-to-YUVr and the RGB-to-YCgCo-R transformations are not orthogonal but bi-orthogonal.

3. A NEW FAMILY OF LOW COMPLEXITY TRANSFORMATIONS

In general, it is assumed that the Karhunen-Loève transform (KLT, also known as principal component analysis) provides the maximum decorrelation. It rotates the coordinate system in the direction of maximal correlation between the RGB values. There are, however, justified reasons not to use the KLT as colour-decorrelation step: (i) it has to be considered that the rounding operations at lifting steps with non-integer coefficients lead to non-linear effects disturbing the optimal rotation of the coordinate system; (ii) the KLT is an orthogonal transformation. It is well-known that bi-orthogonal transformations can perform better, dependent on the application (for example, in wavelet transforms); (iii) the adaptive computation of the KLT and its factorisation into lifting steps is relativly complex; and (iv) in application to image compression, the optimality criterion for the colour transformation is not maximum decorrelation of colour components, but the minimal bitrate of the compressed file.

In the following, we focus on low-complexity transformations, which (i) can be performed using variants of the processing schemes depicted in Figures 1 and 2 and (ii) have the same dynamic range[1].

3.1. Computation of luminance component Y

Both colour transformations discussed above use

$$Y \;\; = \;\; \lfloor (R + 2G + B)/4 \rfloor \qquad (8)$$

as a trade-off between decorrelation performance and low complexity. The computation of Y can be varied by the values of α in the structure of Figure 1 and β in the structure Figure 2.

The least complex variant is simply to copy one component. The green component could be chosen: $Y = G$. This is achieved by setting $\alpha = 0$ or $\beta = 1$, respectively. There is, however, justified reason also to consider R and B as possible Y component: $Y = R$ or $Y = B$.

Now it becomes clear that the ‘best’ colour space we are seeking is not necessarily a decomposition in a luminance and two chrominance components. Nevertheless, we will stick to theses terms for simplicity.

For a majority of natural images, averaging the RGB values is more related to the KLT: $Y = \lfloor (R + G + B)/3 \rfloor$, which is achieved by either using $\alpha = 1/3$ or $\beta = 1/3$. While copying one component reduces the amount of computation significantly, the division by three takes more time than simple bit-shift operations. The proposal in [9] approximates $1/3$ with $3/8$ and $5/16$, converting the division in a multiplication and a shift operation. For this purpose, we have to set $\alpha = 5/16$ or $\beta = 3/8$, respectively, leading to $Y = \lfloor (5R + 6G + 5B)/16 \rfloor$. To limit the number of possible colour spaces, this approximation will not be considered further.

Regarding the modification of Y, there are the following additional transformation matrices

$$\mathbf{A}_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{pmatrix} \quad \text{and} \quad \mathbf{A}_3 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{pmatrix}. \qquad (9)$$

with respect to the processing flow in Figure 1. The structure in Figure 2 can be utilised for

$$\mathbf{C}_2 = \begin{pmatrix} 0 & 1 & 0 \\ -1/2 & 1 & -1/2 \\ 1 & 0 & -1 \end{pmatrix} \text{ and } \mathbf{C}_3 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1/2 & 1 & -1/2 \\ 1 & 0 & -1 \end{pmatrix}. \quad (10)$$

We would like to note that the colour space O1O2O3 [10], which is referred to by some papers, is equivalent to the transformation matrix $\mathbf{C}_3$. The mere difference lies in a scaling of values by minus two leading to an unusual dynamic range of the chrominance values in the O1O2O3 space.

3.2. Computation of chrominances U and V

Considering the colour transformations discussed above, there are only four different formulae to compute a chrominance value: $R - G$, $B - G$, $R - B$, and $G - (R + B)/2$. The sign has no influence on the decorrelation, i.e. $R - G$ and $G - R$ are qualitatively equivalent.

[1] If R, G, and B require eight bits per value each, then Y also requires eight, while U and V require nine bits.

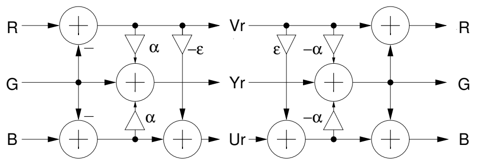

Fig. 3. Structure of the YUVr transformation with an additional lifting step

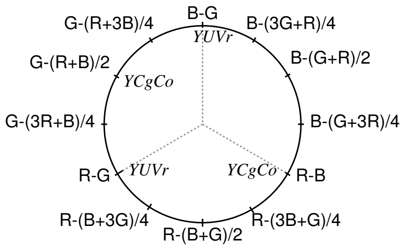

Fig. 4. Circle of possible low-complexity calculations of the chrominance components U and V with gradual changes

The variety of colour spaces can be extended via permutations of the RGB input values modifying (2) and (9) to

$$\mathbf{A}_4 = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \mathbf{A}_5 = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \tag{11}$$

$$\mathbf{A}_6 = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \mathbf{A}_7 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} \tag{12}$$

$$\mathbf{A}_8 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \mathbf{A}_9 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{pmatrix} . \tag{13}$$

The matrices in (7) and (10) can be changed to

$$\mathbf{C}_4 = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ 1 & -1/2 & -1/2 \\ 0 & 1 & -1 \end{pmatrix} \quad \mathbf{C}_5 = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ -1/2 & -1/2 & 1 \\ 1 & -1 & 0 \end{pmatrix} \tag{14}$$

$$\mathbf{C}_6 = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1/2 & -1/2 \\ 0 & 1 & -1 \end{pmatrix} \quad \mathbf{C}_7 = \begin{pmatrix} 0 & 0 & 1 \\ -1/2 & -1/2 & 1 \\ 1 & -1 & 0 \end{pmatrix} \tag{15}$$

$$\mathbf{C}_8 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/2 & -1/2 \\ 0 & 1 & -1 \end{pmatrix} \quad \mathbf{C}_9 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1/2 & -1/2 & 1 \\ 1 & -1 & 0 \end{pmatrix} . \tag{16}$$

3.3. Structure with increased flexibility

The variety of U/V-computations based on permutations is very limited, especially as the computation of Y is dependent on the computation of U and V.

The flexibility can be significantly increased by using only one additional lifting step with coefficient ε, extending the YUVr structure (**Fig. 3**) and enabling many more different computations of the chrominances.

If we stick to low-complexity transformations, the variety, which is depicted in **Figure 4**, can be easily achieved.

Keeping $\alpha = 1/4$ and setting ε equal to $1/4$, $1/2$, or $3/4$ results in

$$\mathbf{E}_1 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ -1/4 & -3/4 & 1 \\ 1 & -1 & 0 \end{pmatrix} \; \mathbf{E}_2 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ -1/2 & -1/2 & 1 \\ 1 & -1 & 0 \end{pmatrix} \tag{17}$$

$$\mathbf{E}_3 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ -3/4 & -1/4 & 1 \\ 1 & -1 & 0 \end{pmatrix} . \tag{18}$$

Exchanging the input values of R and B, R and G, or G and B in $\mathbf{E}_1$ to $\mathbf{E}_3$ yields

$$\mathbf{E}_4 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1 & -3/4 & -1/4 \\ 0 & -1 & 1 \end{pmatrix} \qquad \mathbf{E}_5 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1 & -1/2 & -1/2 \\ 0 & -1 & 1 \end{pmatrix} \tag{19}$$

$$\mathbf{E}_6 = \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1 & -1/4 & -3/4 \\ 0 & -1 & 1 \end{pmatrix} \tag{20}$$

$$\mathbf{E}_7 = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -3/4 & -1/4 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \mathbf{E}_8 = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -1/2 & -1/2 & 1 \\ -1 & 1 & 0 \end{pmatrix} \tag{21}$$

$$\mathbf{E}_9 = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -1/4 & -3/4 & 1 \\ -1 & 1 & 0 \end{pmatrix} \qquad \mathbf{E}_{10} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ -1/4 & 1 & -3/4 \\ 1 & 0 & -1 \end{pmatrix} \tag{22}$$

$$\mathbf{E}_{11} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ -1/2 & 1 & -1/2 \\ 1 & 0 & -1 \end{pmatrix} \quad \mathbf{E}_{12} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ -3/4 & 1 & -1/4 \\ 1 & 0 & -1 \end{pmatrix} . \tag{23}$$

Again, it is possible to exchange an additional pair of columns in (21) to (23) leading to

$$\mathbf{E}_{13} = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -3/4 & 1 & -1/4 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathbf{E}_{14} = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -1/2 & 1 & -1/2 \\ -1 & 0 & 1 \end{pmatrix} \tag{24}$$

$$\mathbf{E}_{15} = \begin{pmatrix} 1/2 & 1/4 & 1/4 \\ -1/4 & 1 & -3/4 \\ -1 & 0 & 1 \end{pmatrix} \quad \mathbf{E}_{16} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ 1 & -1/4 & -3/4 \\ 0 & 1 & -1 \end{pmatrix} \tag{25}$$

$$\mathbf{E}_{17} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ 1 & -1/2 & -1/2 \\ 0 & 1 & -1 \end{pmatrix} \quad \mathbf{E}_{18} = \begin{pmatrix} 1/4 & 1/4 & 1/2 \\ 1 & -3/4 & -1/4 \\ 0 & 1 & -1 \end{pmatrix} . \tag{26}$$

With $\alpha = 0$ and variation of ε from $1/4$ to $1/2$ to $3/4$, we get matrices $\mathbf{D}_1$ to $\mathbf{D}_{18}$, differing from $\mathbf{E}_1$ to $\mathbf{E}_{18}$ only in the first line, which is responsible for the computation of Y. Here, $1/2$ is substituted by '1' and $1/4$ by '0'. It has to be mentioned that, in contrast to the exchange of the RGB values, the assignment of U and V (i.e. the order of the computed chrominances) has no impact on the compression of the decorrelated colour image data as long as the components U and V are processed in identical manner in the coding stage.

Finally, we have to consider the cases where Y is the average of R, G, and B ($\alpha = 1/3$). Here, only $\varepsilon = 1/4$ is of interest, since choosing $\varepsilon = 1/2$ repeats matrices from the $\mathbf{C}_i$ family branch and $\varepsilon = 3/4$ is achieved via $\varepsilon = 1/4$ plus permutation of the RGB values

$$\mathbf{F}_1 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1/4 & -3/4 & 1 \\ 1 & -1 & 0 \end{pmatrix} \mathbf{F}_2 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -3/4 & -1/4 \\ 0 & -1 & 1 \end{pmatrix} \tag{27}$$

$$\mathbf{F}_3 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -3/4 & -1/4 & 1 \\ -1 & 1 & 0 \end{pmatrix} \mathbf{F}_4 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -3/4 & 1 & -1/4 \\ -1 & 0 & 1 \end{pmatrix} \tag{28}$$

$$\mathbf{F}_5 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ -1/4 & 1 & -3/4 \\ 1 & 0 & -1 \end{pmatrix} \mathbf{F}_6 = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1 & -1/4 & -3/4 \\ 0 & 1 & -1 \end{pmatrix} . \tag{29}$$

Sixty colour transformations can be distinguished in total: nine matrices $\mathbf{A}_i$ related to the processing structure in Figure 1, nine matrices $\mathbf{C}_i$ related to the processing structure in Figure 2, eighteen matrices each for $\mathbf{D}_i$ and $\mathbf{E}_i$ as well as six matrices $\mathbf{F}_i$, all related to the extended processing structure in Figure 3. The identity $(Y\,U\,V) = (R\,G\,B)$ (i.e. no colour transformation) has to be considered as well.

On account of the selected low-complex processing strategies, not all calculations depicted in Figure 4 can be combined. At least one chrominance component is equal to the simple difference between two of the RGB values. The second chrominance component lies on the circumference of the opposite sector.

4. AUTOMATIC SELECTION OF SUITABLE COLOUR TRANSFORMATIONS

The minimum bitrate which can be obtained by a compression system depends on the mean information content of the original data. So, it is logical to examine the entropies of the three components Y, U, and V after the colour transformation. The assumption is: the smaller the sum entropy

$$H_{sum} = H(Y) + H(U) + H(V)\,, \tag{30}$$

the lower the bitrate in the compressed signal.

Since many colour spaces share the same Y, U, or V component, each component has to be computed only once. The computational overhead of the adaptive selection can be limited further by using only a subset of pixels for the examination. Investigations have shown that, in general, 10^4 pixels are enough, if they are spread over the whole image.

The only problem with the method described above is that the colour transform can propagate noise from one colour component into another, disturbing the subsequent step of spatial decorrelation. We have implemented a simple prediction step, accounting for this effect.

Let $x[n, m]$ be a certain signal value at row $0 \leq n < height$ and column $0 < m < width$ in either the Y, U, or V component, then the prediction error is computed as

$$e[n, m] = x[n, m] - x[n, m-1] \quad \forall\, n, m \tag{31}$$

and the entropies are determined based on these prediction errors

$$H(E)_{sum} = H(E_Y) + H(E_U) + H(E_V)\;. \tag{32}$$

So, the colour space leading to the smallest $H(E)_{sum}$ is chosen. The selection can be transmitted to the decoder with only five bits overhead.

Table 1. Results in bits per pixel using different colour-space settings. See text for details.

colour	averaged [bpp]			
	all 219 images		only photos	
space	LOCO	JPEG2K	LOCO	JPEG2K
RGB	10.015	10.855	11.673	12.016
YCgCo-R	8.154	8.767	9.287	9.553
$\mathbf{E}_2$	8.116	8.784	9.198	9.524
$\mathbf{A}_2$	8.115	8.765	9.262	9.580
$\mathbf{D}_1$	8.102	8.783	9.219	9.557
$\mathbf{E}_1$	8.093	8.755	**9.184**	**9.509**
YUVr	8.091	8.738	9.227	9.532
best	7.858	8.521	9.073	9.411
automatic	7.876	8.557	9.090	9.430
automatic (10^4)	7.877	8.559	9.090	9.431

5. INVESTIGATIONS

The examination and comparison of 61 colour spaces require an adequate number of test images with diverse characteristics. As a compromise between statistical relevance and computational time, a set of 219 images was assorted [11]. The images are taken from different sources (internet or standardisation groups), avoiding results which are biased to a particular image generation system. Photos account for 154 images, with the remaining images being computer-generated images or of mixed content. Two different compression algorithms are used as benchmark software: the LOCO-I algorithm [12], which combines an adaptive spatial prediction step with context-based rice coding, and a JPEG2000-like coding algorithm based on integer wavelet transformation and block-based arithmetic coding of bit-planes.

Each colour image was encoded 61 times using the RGB colour space or different colour transformations as described above. The average bitrates are listed in **Table 1**. In rows RGB to YUVr, it shows the averaged bitrates if the colour transformation is fixed for all images. It clearly can be seen that the YUVr space outperforms RGB and YCgCo-R on average for this particular set of images. Colour space $\mathbf{A}_2$ is included, since the corresponding transform has the lowest complexity. If only photos are considered, the ranking of the colour spaces is slightly different. The new colour space $\mathbf{E}_1$ now leads to the lowest bitrate.

For photos only, the colour space $\mathbf{D}_1$ is more suitable than YUVr when using the LOCO-I algorithm despite its somewhat lower complexity ($V = R - G$, $U' = B - G$, $Y = G$, $U = U' - \lfloor V/4 \rfloor$). In general the result of $\mathbf{D}_1$ is closer to the result of $\mathbf{E}_1$, when using the LOCO-I. As $\mathbf{D}_1$ and $\mathbf{E}_1$ differ only in Y, it seems that the spatial prediction step in LOCO-I is more sensitive to the propagated colour noise than the integer wavelet transform. This sensitivity reduces the positive effect of computing Y based on all three components.

Since all images were compressed 61 times using different colour spaces, we can take for each image the run leading to the lowest bitrate in bits per pixel. The colour space used for this run determines the ‘best’ colour transformation. In Table 1, the row ‘best’ shows the result after averaging all smallest bpp values of the images. It is distinctly smaller than the results with any fixed colour space. Choosing an appropriate colour space can obviously improve the compression performance. **Figure 5** shows how often a particular colour space was the best one. It can be seen that the selection differs slightly depending on the compression algorithm. The highest difference has YCgCo. It is best in ten cases when using the wavelet-based coding scheme, but, in combination with

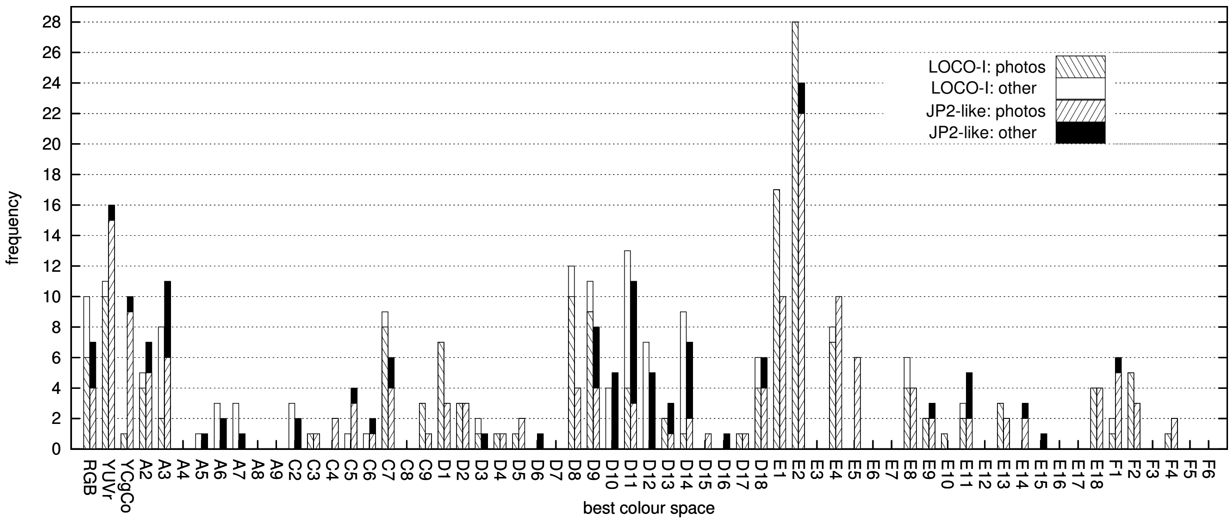

Fig. 5. Histogram of best colour spaces for 219 test images, separated into photos and other images

LOCO-I, it is best only for a single image. Furthermore, it can be observed that some colour spaces are more suitable for natural images (photos) and some (e.g. $\mathbf{D}_{10}$ – $\mathbf{D}_{14}$) are more applicable to synthetic images.

Using the automatic selection as described above, the results in line 'automatic' can be obtained (Tab. 1). They are considerably better than any fixed colour space and very close to the theoretically possible (line 'best'). Performing the selection based on only 10^4 pairs of pixels (enabling the left-neighbour prediction) marginally increases the bitrates on average.

6. SUMMARY AND CONCLUSION

Based on a wide variety of new low-complexity colour transforms, it could be shown that the compression results improve if the colour transformation is selected according to the image content. The analysis has revealed that the YUVr colour space is more suitable than the YCgCo-R colour space for a broad range of images. The transforms based on the new matrices $\mathbf{E}_1$ and $\mathbf{E}_2$, however, are more often selected if the image is a photo (Fig. 5). With respect to the investigated photos, the new colour space $\mathbf{E}_1$ outperforms YUVr. The calculations of the RGB-to-$\mathbf{E}_1$ transform are

$$\begin{aligned} V &= R - G & U' &= B - G \\ Y &= G + \lfloor (U' + V)/4 \rfloor & U &= U' - \lfloor V/4 \rfloor \ . \end{aligned} \tag{33}$$

Synthetic images mostly compress best with colour spaces using R, G or B as Y component.

Without significant loss of performance, a suitable colour space can be selected based on a simple technique examining the entropies of prediction errors in the luminance and chrominance components. Generally, the computational overhead is marginal, as the image analysis can be restricted to only 10^4 pairs of pixel.

Future investigations will show whether the new colour spaces also increase the coding gain in lossy compression.

7. REFERENCES

[1] Fukuma, S.; Iwahashi, M.; Kambayashi, N.: Lossless color coordinate transform for lossless color image coding. *Proc. of IEEE APCCAS*, 24-27 Nov 1998, 595 – 598

[2] ISO/IEC JTC1/SC29/WG11 N1890, *Information technology – JPEG 2000 Image Coding System.* JPEG 2000 Part I, Final Draft Intern. Standard 15444, 25 Sep. 2000

[3] Malvar, H.; Sullivan, G.: YCoCg-R: A color space with RGB reversibility and low dynamic range. *ISO/IEC JTC1/SC29/WG11*, Document JVT-I014, 2003

[4] Hao, P.; Shi, Q.: Comparative study of color transforms for image coding and derivation of integer reversible color transform. *Proc. of Int. Conf. Pattern Recogn.*, Sept. 2000, Vol.3, 224–227

[5] Han, S.-E.; Tao, B.; Cooper, T.; Tastle, I.: Comparison between Different Color Transformations for the JPEG 2000. *Proc. of PICS 2000*, Portland, OR, March 2000, 259–263

[6] Marpe, D.; Kirchhoffer, H.; George, V.; Kauff, P.; Wiegand, T.: An Adaptive Color Transform Approach and its Application in 4:4:4 Video Coding. *Proc. of EUSIPCO 2006*, Florence, Italy, Sept. 2006

[7] Assche, S. van; Philips, W.; Lemahieu, I.: Lossless compression of pre-press images using a novel colour decorrelation technique. *Pattern Recognition*, Vol.32, No.3, March 1999, 435–441

[8] Pasteau, F.; Strauss, C.; Babel, M.; Déforges, O.; Bédat, L.: Adaptive Color Decorrelation for Predictive Image Codecs. *EUSIPCO 2011*, Barcelona, Spain, 2011, 1100–1104

[9] Topiwala, P.; Tu, C.: New Invertible Integer Color Transforms Based on Lifting Steps and Coding of 4:4:4 Video. *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6* Document JVT-I015r3, San Diego, Sept. 2003

[10] Nakachi, T.; Fujii, T.; Suzuki, J.: Lossless and near-lossless compression of still color images. *Proc. of ICIP 1999*, Vol.1, 453–457

[11] www1.hft-leipzig.de/strutz/Papers/Testimages/CT/ last visited 16.12.2011

[12] Weinberger, M.J.; Seroussi, G.; Sapiro, G.: LOCO-I: A Low Complexity, Context Based, Lossless Image Compression Algorithm. *Proc. of DCC 1996*, 140–149