Multiplierless Reversible Colour Transforms and their Automatic Selection for Image Data Compression

Tilo Strutz

***Abstract*—The efficient compression of colour images requires a step which takes the dependencies between the colour components into account. This is mostly realised with a colour transformation mapping the red, green and blue components into another representation. The transformation must be reversible if the compression system is to support the lossless reconstruction of the images.**

In this paper, we propose an entire family of multiplierless reversible colour transforms and investigate their performance in lossless image compression. When using the LOCO-I algorithm or lossless JPEG2000 for compression, results show that the compression performance (in terms of bits per pixel) using the optimal colour-space selection leads to a distinct reduction of the mean bitrate compared to any fixed colour space. Further, it is shown that it is possible to automatically select a suitable colour space without significant loss in compression efficiency compared to the optimal selection. This automatic selection improves the compression performance independently of whether LOCO-I or JPEG2000 is applied.

***Index Terms*—image compression, reversible colour transformation**

I. Introduction

The reversible compression of images 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 $\mathbf{A}$

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

The elements of the matrix should be chosen so that the compression of the image leads to a minimum bitrate.

In [1], it was stated that the implementation with ladder networks (also known as lifting scheme) 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.

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 [3] as proposed for the fidelity range extension of the video coding standard H.264/AVC [4].

This paper develops a new family of reversible low-complexity colour transformation, which can be computed based on additions and shift operations only, while covering a broad range of distinct colour spaces. The improvements in terms of bitrate reduction are investigated. Further, it will be shown that it is possible to select a suitable colour space automatically with reasonable costs. The paper is based in part on an earlier work [5]. The theoretical basis of the colour transforms, however, has been simplified and the investigations have been significantly extended in several ways.

The image-adaptive decorrelation of the colour components has already been addressed in literature. In [6], eleven transformations were compared with the focus on lossy compression, and an integer reversible transform was derived from a three-channel discrete cosine transform. This transform, however, performed worse than YUVr in lossless compression. The YUVr colour space was compared with human vision-based colour spaces and the Karhunen-Loève transform (KLT) in [7]. This investigation solely covered lossy compression. In application to high-quality video compression, the adaptive selection between YUVr, YCgCo-R and a third colour space was successfully investigated in [8] based on a block-by-block basis. The colour decorrelation was applied after the spatial decorrelation, making the switching between colour spaces possible without degradation of the spatial prediction. A block-based approach was also proposed in [9]. Here, the adaptation was performed with a local re-computation of the KLT and a relatively complicated quantisation of the matrix coefficients to make the transformation reversible. The highest degree of adaptation is achieved on a pixel-by-pixel basis. In [10], the dependencies between the colour components are utilised for improving the prediction values. The adaptation is possible without transmitting additional side information. The integration of the colour decorrelation into the prediction step was also investigated in [11].

It seems to be obvious that local adaptation of processing steps is most likely superior to global adaptation. Unfortunately, the switching between different colour spaces introduces non-linearity, influencing the subsequent processing. For this reason, this paper concentrates in the beginning 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 without influencing too

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

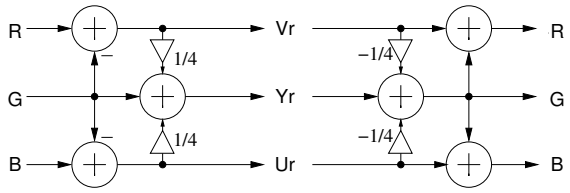

Fig. 1. Processing steps of the reversible colour transformation (YUVr) defined in JPEG2000

many coding details. The proposed selection process takes the problem into account which stems from the uncorrelated noise over the three colour components, see [8]. Later on, the block-wise selection of colour spaces will also be discussed.

Two different compression algorithms are used as benchmark software. In the first part of investigations, the LOCO-I algorithm [12] is applied, which combines an adaptive spatial prediction step with a context-based Rice code. These results are later compared to the compression with the lossless mode of JPEG2000 [13], which combines integer wavelet transformation and block-based arithmetic coding of bit-planes.

The paper is organised as follows: Section II reviews the reversible colour transform used in JPEG2000. Based on this, Section III derives a new family of related colour transformations. Section IV discusses the automatic selection of a suitable colour space and presents first results. Section V evaluates the influence of the applied compression scheme on the effectiveness of the automated colour-space selection. Section VI discusses the results and concludes the paper.

II. Review of the JPEG2000 reversible colour transformation

The reversible colour transformation defined in JPEG2000 (YUVr) uses the matrix

$$\mathbf{A} = \begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{pmatrix}. \tag{2}$$

Together with the permutation matrix

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

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

$$\begin{aligned} \mathbf{A} &= \mathbf{P}_1 \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.25 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0.25 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \mathbf{P}_1 \times \begin{pmatrix} 1 & 0 & 0 \\ 0.25 & 1 & 0.25 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}. \end{aligned} \tag{4}$$

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 is

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = (\mathbf{A}_1)^{-1} \cdot \begin{pmatrix} Y_\mathrm{r} \\ U_\mathrm{r} \\ V_\mathrm{r} \end{pmatrix} \tag{5}$$

with

$$\begin{aligned} (\mathbf{A}_1)^{-1} &= \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ -0.25 & 1 & -0.25 \\ 0 & 0 & 1 \end{pmatrix} \times \mathbf{P}_1^\mathrm{T} \\ &= \begin{pmatrix} 1 & -0.25 & 0.75 \\ 1 & -0.25 & -0.25 \\ 1 & 0.75 & -0.25 \end{pmatrix}. \end{aligned} \tag{6}$$

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

$$U_\mathrm{r} = B - G \quad V_\mathrm{r} = R - G \quad Y = G + \left\lfloor \frac{U_\mathrm{r} + V_\mathrm{r}}{4} \right\rfloor \tag{7}$$

$$G = Y - \left\lfloor \frac{U_\mathrm{r} + V_\mathrm{r}}{4} \right\rfloor \quad R = V_\mathrm{r} + G \quad B = U_\mathrm{r} + G\,. \tag{8}$$

The operator $\lfloor x \rfloor$ rounds x downwards to the nearest integer value.

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

Note that the RGB-to-YUVr transformation is not orthogonal but bi-orthogonal.

III. Derivation of a 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 (e.g. wavelet transforms); (iii) the adaptive computation of the KLT and its factorisation into lifting steps is relatively complex; and most important (iv) in application to image compression, the optimality criterion for the colour transformation is not maximum decorrelation of colour components, but the minimal size 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 Figure 1, (ii) are multiplierless and

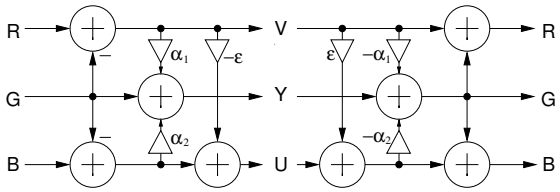

Fig. 2. Extension of the YUVr transformation by an additional lifting coefficient ε

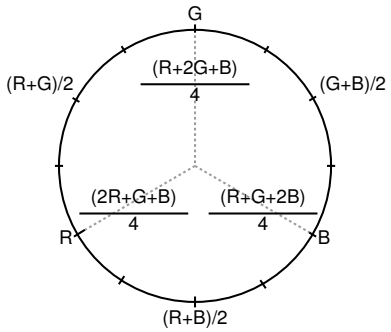

Fig. 3. Investigated computations of the Y component

(iii) have the same dynamic range[1]. Despite this restriction, the space of all possible colour transformations is sampled on such a fine grid that it is not worth considering more complex computations.

Using two simple modifications, the YUVr transform can be converted into a very flexible structure. These modifications are: separate coefficients α_1 and α_2 (instead of the fixed value of 1/4) and an additional lifting step extending the YUVr structure (**Fig. 2**). The variation of α_1, α_2 and ε enables different computations of the Y, U and V components. **Figure 3** and **Figure 4** depict all cases, which are accessible via low-complexity multiplierless transforms, while covering a sufficient broad and densely sampled range.

A. Computation of luminance component Y

The JPEG2000 reversible colour transformations discussed above uses

$$Y = \lfloor (R + 2G + B)/4 \rfloor \tag{9}$$

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

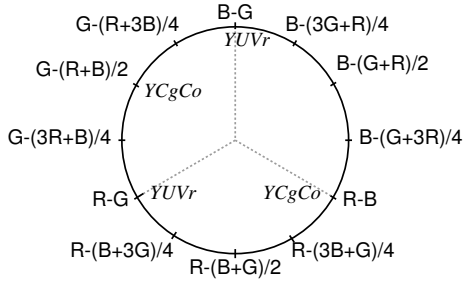

Fig. 4. Circle of possible low-complexity calculations of the chrominance components U and V without multiplications

TABLE I
COMPUTATION OF COMPONENT Y DEPENDENT ON THE LIFTING COEFFICIENTS α_1 AND α_2

i	α_1	α_2	Y
1	0	0	G
2	1	0	R
3	0	1	B
4	1/2	0	$\lfloor (G+R)/2 \rfloor$
5	0	1/2	$\lfloor (G+B)/2 \rfloor$
6	1/2	1/2	$\lfloor (R+B)/2 \rfloor$
7	1/4	1/4	$\lfloor (R+2G+B)/4 \rfloor$
8	1/2	1/4	$\lfloor (2R+G+B)/4 \rfloor$
9	1/4	1/2	$\lfloor (R+G+2B)/4 \rfloor$

as a trade-off between decorrelation performance and low complexity. The computation of Y can be varied by the values of α_1 and α_2 in the structure of Figure 2.

The least complex variant is simply to copy the green component with $\alpha_1 = \alpha_2 = 0$.

$$Y = G. \tag{10}$$

The usage of two different coefficients α_1 and α_2 extends the variety of multiplierless computations compared to the family of transforms presented in [5]. In addition, the computations of U and V become independent of Y. In total, we concentrate on nine combinations (**Tab. I**).

Now it becomes obvious that the 'best' colour space we are looking for is not necessarily a decomposition in a luminance and two chrominance components. Nevertheless, we will stick to these terms for simplicity.

B. Computation of chrominances U and V

In the YUVr transform (eq.2), the V component is computed with $V = R - G$. This can be changed with the processing structure in Figure 2 only if the input values RGB are permuted. Starting with $\varepsilon = 0$, there are three unique computations

j	ε	V	U	permutation
1	0	$R-G$	$B-G$	
2	0	$G-R$	$B-R$	$R \leftrightarrow G$
3	0	$R-B$	$G-B$	$B \leftrightarrow G$

It has to be mentioned that the permutation of R, G and B also influences the computation of Y. If Y should stay fixed, the coefficients α_1 and α_2 must be changed accordingly.

Setting $\varepsilon = 1/4$, the computation of U changes, while V still depends only on the permutation:

j	ε	V	U	permutation
4	1/4	$R-G$	$B-\lfloor (R+3G)/4 \rfloor$	
5	1/4	$G-R$	$B-\lfloor (G+3R)/4 \rfloor$	$R \leftrightarrow G$
6	1/4	$R-B$	$G-\lfloor (R+3B)/4 \rfloor$	$B \leftrightarrow G$
7	1/4	$B-G$	$R-\lfloor (B+3G)/4 \rfloor$	$R \leftrightarrow B$
8	1/4	$G-B$	$R-\lfloor (G+3B)/4 \rfloor$	$R \rightarrow G \rightarrow B \rightarrow R$
9	1/4	$B-R$	$G-\lfloor (B+3R)/4 \rfloor$	$R \rightarrow B \rightarrow G \rightarrow R$

Setting $\varepsilon = 1/2$ leads to three additional variants:

j	ε	V	U	permutation
10	1/2	$R-G$	$B-\lfloor (R+G)/2 \rfloor$	
11	1/2	$R-B$	$G-\lfloor (R+B)/2 \rfloor$	$B \leftrightarrow G$
12	1/2	$B-G$	$R-\lfloor (B+G)/2 \rfloor$	$R \leftrightarrow B$

All other permutations of R, G and B repeat one of the computations above.

Since Y can be independently computed from the chrominances, there are $9 \times 12 = 108$ different combinations. While the exchange of the RGB input values modifies the computation, 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 independently processed in identical manner in the coding stage.

Let $\mathbf{A}_{i,j}$ be a particular colour-transform matrix, combining a Y component i and a pair of chrominances j. The factorisation of, for example,

$$\mathbf{A}_{4,10} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ -1/2 & -1/2 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

is

$$\mathbf{A}_{4,10} = \mathbf{P}_1 \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/2 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} . \quad (11)$$

The most-right matrix shows that R, G, and B are processed without permutation. The coefficients are $\alpha_1 = 1/2$, $\alpha_2 = 0$ and $\varepsilon = 1/2$. The computation based on the flow chart in Figure 2 is, therefore,

$$\begin{aligned} V &= R - G & U' &= B - G \\ Y &= G + \lfloor V/2 \rfloor & U &= U' - \lfloor V/2 \rfloor \ . \end{aligned} \quad (12)$$

The colour transform $\mathbf{A}_{7,7}$, in contrast, is based on an exchange of R and B, $\alpha_1 = \alpha_2 = 1/4$, and $\varepsilon = 1/4$

$$\begin{aligned} \mathbf{A}_{7,7} &= \begin{pmatrix} 1/4 & 1/2 & 1/4 \\ 1 & -3/4 & -1/4 \\ 0 & -1 & 1 \end{pmatrix} \\ &= \mathbf{P}_1 \times \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/4 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 1/4 \\ 0 & 0 & 1 \end{pmatrix} \\ &\quad \times \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \times \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} . \end{aligned} \quad (13)$$

The computation would be

$$\begin{aligned} V &= B - G & U' &= R - G \\ Y &= G + \lfloor (V + U')/4 \rfloor & U &= U' - \lfloor V/4 \rfloor \ . \end{aligned} \quad (14)$$

Note that the matrix $\mathbf{A}_{7,1}$ corresponds to YUVr (JPEG2000) and $\mathbf{A}_{7,11}$ corresponds to YCgCo-R described in [3].

C. Y_1Y_2C colour spaces for data with low correlation

The colour spaces discussed above utilise one component Y containing averaged or unchanged data and two components U and V containing difference signals. The RGB colour space has no difference signals.

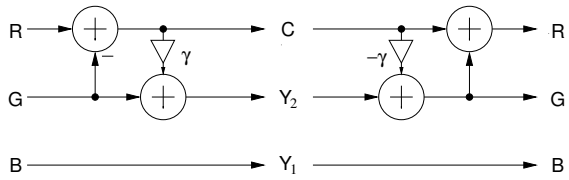

Fig. 5. Colour-transformation structure for images having components with low correlation

TABLE II
Y_1Y_2C COLOUR TRANSFORMATIONS FOR IMAGE COMPONENTS WITH ONLY LOW CORRELATION

l	γ	Y_1	Y_2	C	permutation
1	0	B	G	$R-G$	
2	0	R	G	$B-G$	$B \leftrightarrow R$
3	0	B	R	$G-R$	$R \leftrightarrow G$
4	0	G	R	$B-R$	$R \rightarrow B \rightarrow G \rightarrow R$
5	0	R	B	$G-B$	$R \rightarrow G \rightarrow B \rightarrow R$
6	0	G	B	$R-B$	$B \leftrightarrow G$
7	1/2	B	$\lfloor (R+G)/2 \rfloor$	$R-G$	
8	1/2	R	$\lfloor (B+G)/2 \rfloor$	$B-G$	$B \leftrightarrow R$
9	1/2	G	$\lfloor (R+B)/2 \rfloor$	$R-B$	$B \leftrightarrow G$

For images with two correlated components and a third component, which is neither correlated with the first nor the second component, it might be favourable to use a colour space having only one difference signal, while the others both contain either unprocessed signal values or averaged values. **Figure 5** shows a suitable set-up. If γ is equal to 0, then the two components G and B remain unchanged, while the third component is $R-G$. B is assumed as having no or only low correlation with R and G. The permutation of the input values leads to six different combinations. Since the difference signal C is computed by two components which are correlated, it does make sense to average these two components forming Y_2. For the sake of simplicity, we only consider $\gamma = 1/2$. In total, there are nine additional colour transforms (**Tab. II**). The corresponding transformation matrices are called $\mathbf{B}_l$.

D. Complexity of colour transforms

According to the processing structures in Figure 2, the operations reduce to additions and bitwise shifts, as long as the coefficients α_1, α_2, and ε are powers of two or are equal to zero. **Table III** compares selected colour transformations. Again, one has to keep in mind that the index i is assigned to the selected computation of Y (see Tab.I). The choice of the coefficients α_1 and α_2 depends on the permutation of the input RGB values. If ε is equal to 1/2 or equal to 1/4 (matrices $\mathbf{A}_{i,4\ldots12}$), then we have to add one shift operation and one addition.

Using a different processing structure (**Figure 6**), which is inspired by the computation of the YCgCo-R transform [3], the complexity can be reduced for some of the transforms. Here, the computation of Y depends on both, β_1 and β_2 and it cannot be treated separately from the chrominances. There are six cases of interest, numbered with k (**Table IV**).

Let $\mathbf{C}_k$ be a colour transform matrix based on the processing structure in Figure 6 computing the components Y, U and V as given in the tables above. **Table V** shows the corresponding

TABLE III
COMPARISON OF SELECTED COLOUR TRANSFORMS IN TERMS OF COMPUTATIONAL COSTS

colour space	adds	shifts
RGB	0	0
$B_1, \ldots B_6$	1	0
$B_7, \ldots B_9$	1	1
$A_{1,1}$, $A_{2,2}$, $A_{3,3}$	2	0
$A_{2,1}$, $A_{1,2}$, $A_{2,3}$	3	0
$A_{3,1}$, $A_{3,2}$, $A_{1,3}$	3	0
$A_{4,1}$, $A_{4,2}$, $A_{6,3}$	3	1
$A_{5,1}$, $A_{6,2}$, $A_{5,3}$	3	1
$A_{6,1}$, $A_{5,2}$, $A_{4,3}$	4	1
$A_{7,1}$, $A_{8,2}$, $A_{9,3}$	4	1
$A_{8,1}$, $A_{7,2}$, $A_{8,3}$	4	2
$A_{9,1}$, $A_{9,2}$, $A_{7,3}$	4	2

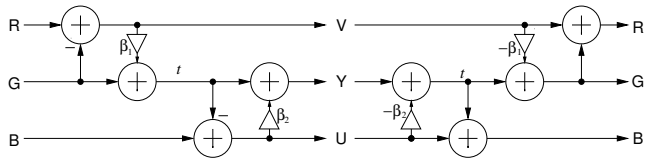

Fig. 6. Alternative processing structure for colour transformations

numbers of required additions and shift operations and gives a comparison to equivalent $\mathbf{A}_{i,j}$ matrices. $\mathbf{C}_2$ corresponds to the YCgCo-R transform.

E. Investigations

The examination and comparison of 118 colour spaces, including RGB, requires an adequate number of test images with diverse characteristics. As a compromise between statistical relevance and computational time, a set of 746 images was evaluated [14]. The images are taken from different sources reducing the influence of a particular image generation system. Photos account for 499 images, with the remaining images being computer-generated images or of mixed content.

The LOCO-I compression algorithm [12] was used as benchmark software, since it is the core algorithm of the international standard for lossless image compression JPEG-LS. It combines an adaptive spatial prediction step with context-based rice coding. As compression schemes using spatial decorrelation like prediction are not optimal for the processing of synthetic images, the investigations differentiate between the entire set of images and a set only containing the 499 photographs.

Each colour image was encoded 118 times using the RGB colour space or different colour transformations as described above.

If we 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. **Table VI** shows how often a particular colour space led to the lowest bitrate. The rows of the table are sorted according to V and U. The main part of the table contains the frequencies of the colour spaces introduced in Subsections III-A and III-B. This is extended by two lines for the colour spaces discussed in Subsection III-C and the RGB colour space. The distribution allows following interpretation

- All variants of computation (luminance $i = 1, 2, \ldots, 9$ and chrominance $j = 1, 2, \ldots, 12$) are selected at least once with the exception of $j = 8$.
- There are combinations of luminance and chrominance computation never leading to the lowest bitrate. This, however, could be caused by the limited number of investigated images.
- Images, which are not photographs, compress best in general, when one of the components R, G, or B is directly mapped to Y.
- Thirteen images (two photos, eleven other) compress best in RGB colour space.
- The Y_1Y_2C colour spaces are especially helpful for synthetic images (22 cases).
- The YUVr colour space ($\mathbf{A}_{7,1}$) was selected only 17 times, YCgCo-R ($\mathbf{A}_{7,11}$) only four times.
- The most effective colour spaces for the photos in this test set are $\mathbf{A}_{7,10}$ ($84\times$) and $\mathbf{A}_{4,10}$ ($54\times$) followed by $\mathbf{A}_{3,10}$ ($44\times$) and $\mathbf{A}_{7,4}$ ($32\times$).
- The colour space $\mathbf{A}_{4,2}$ was also chosen 32 times. Inspecting the images, it turned out that these images contain the same motif with only slight modifications.

The lines 'RGB' to '$\mathbf{A}_{7,4}$' in **Table VII** show the averaged bitrates if the colour transformation is fixed for all images. It can be seen that using any of the listed colour spaces significantly reduces the bitrate compared to the RGB colour space. $\mathbf{A}_{7,4}$ has the best performance closely followed by $\mathbf{A}_{7,10}$, while YCgCo-R is worse than all other listed colour spaces. If we take only photographs into account, i.e. ignoring the results for synthetic and other images, the newly proposed colour space $\mathbf{A}_{7,10}$ leads to the lowest bitrate. The calculations of the RGB-to-$\mathbf{A}_{7,10}$ transform are

$$\begin{aligned} V &= R - G & U' &= B - G \\ Y &= G + \lfloor (V + U')/4 \rfloor & U &= U' - \lfloor V/2 \rfloor \ . \end{aligned} \tag{15}$$

The inverse transform is simply

$$\begin{aligned} U' &= U + \lfloor V/2 \rfloor & G &= Y - \lfloor (V + U')/4 \rfloor \\ B &= U' + G & R &= V + G \ . \end{aligned} \tag{16}$$

The results for $\mathbf{A}_{1,1}$ are included, because it belongs to the $\mathbf{A}_{i,j}$ transforms with lowest complexity.

The line 'best' in Table VII indicates the result when the colour space is chosen which leads to the lowest bitrate for each image. This optimal selection significantly drops the bitrate by 0.246 bpp compared to YUVr, when considering all 746 images, and by 0.173 bpp, if only the photos are take into account.

The next section addresses the problem of automatic selection of a suitable colour space.

IV. AUTOMATIC SELECTION OF SUITABLE COLOUR TRANSFORMATIONS

A. Automatic selection based on entropy

The minimum bitrate which can be obtained by a compression system depends on the mean information content of the original data. So, it seems logical to examine the

TABLE IV
ALTERNATIVE COMPUTATION WITH LOWER COMPLEXITY BASED ON FIGURE 6

k	β_1	β_2	Y	V	U	permutation
1	1/2	1/2	$\lfloor (R+G+2B)/4 \rfloor$	$R-G$	$B-\lfloor (R+G)/2 \rfloor$	
2	1/2	1/2	$\lfloor (R+2G+B)/4 \rfloor$	$R-B$	$G-\lfloor (R+B)/2 \rfloor$	$B \leftrightarrow G$
3	1/2	1/2	$\lfloor (2R+G+B)/4 \rfloor$	$B-G$	$R-\lfloor (G+B)/2 \rfloor$	$B \leftrightarrow R$
4	1/2	0	$\lfloor (R+G)/2 \rfloor$	$R-G$	$B-\lfloor (R+G)/2 \rfloor$	
5	1/2	0	$\lfloor (R+B)/2 \rfloor$	$R-B$	$G-\lfloor (R+B)/2 \rfloor$	$B \leftrightarrow G$
6	1/2	0	$\lfloor (G+B)/2 \rfloor$	$B-G$	$R-\lfloor (G+B)/2 \rfloor$	$B \leftrightarrow R$

TABLE V
COMPLEXITY REDUCTION USING THE ALTERNATIVE PROCESSING STRUCTURE

C_k	adds	shifts	$A_{i,j}$	adds	shifts
C_1	4	2	$A_{9,10}$	5	3
C_2	4	2	$A_{7,11}$	5	3
C_3	4	2	$A_{8,12}$	5	3
C_4	3	1	$A_{4,10}$	4	2
C_5	3	1	$A_{6,11}$	4	2
C_6	3	1	$A_{5,12}$	4	2

entropies of the three components Y, U, and V after the colour transformation. The assumption is: the smaller the sum entropy

$$H_{sum} = H(Y) + H(U) + H(V) \tag{17}$$

the lower the bitrate in the compressed signal. Therefore, a routine was implemented, which computes all possible luminances and chrominances and compares the sum entropies for all possible combinations. Note that it is not necessary to perform all colour transforms separately. Since many colour spaces share the same Y, U, or V component, each component has to be computed only once. In total, merely nine Y components and twelve different U/V components must be tested. The algorithm selects the Y component showing the lowest entropy and the U/V combination with the lowest sum $H(U)+H(V)$. The sum entropy H_{sum} of this colour space has finally to be compared against the entropies of the transforms $\mathbf{B}_1 - \mathbf{B}_9$ and RGB.

According to the described procedure, all images can be encoded using the adaptively chosen colour space. The selection is signalled to the decoder using a single byte, however, seven bits would be enough to distinguish between 118 colour spaces plus greyscale images. The resulting average bitrates are listed in Table VII in line 'automatic'. The result is considerably better on average for all images than any fixed colour space, but it is still clearly worse than the theoretical possible (line 'best'). When only inspecting the photos, it becomes apparent that this selection method fails.

B. Automatic colour-space selection including prediction

The problem of the set-up above is that it takes into account only the decorrelation between the colour components. The colour transform, however, can propagate noise from one colour component into another, disturbing the subsequent step of spatial decorrelation.

Ideally, the image analysis should perform the same spatial decorrelation as the compression system, accounting for this effect. In order to limit the computational efforts, we have implemented a simple left-neighbour prediction.

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

$$e[n,m] = x[n,m] - x[n,m-1] \quad \forall\, n,m \tag{18}$$

and the entropies are determined based on these prediction errors

$$H(E)_{sum} = H(E_Y) + H(E_U) + H(E_V)\ . \tag{19}$$

Chosing the right colour transformation in this manner leads to distinctly better results, see Table VII, line 'automatic (incl. prediction)', which are close to the optimal selection. However, the automatic selection leads only for 282 of the 746 images to the 'best' colour space. Obviously, there is a colour space for most images, leading to a similar bitrate as the best one. The inspection of the results confirms that the two best colour spaces mostly differ only in the computation of one of the three components.

C. Selection based on a reduced set of pixels

The computational costs of adaptivity can be significantly reduced by computing the entropies (as well as the prediction errors) based on a reduced set of pixels. Ideally, this subset should be randomly drawn from the entire set of pixels. For the sake of simplification, we have implemented a procedure which addresses the pixels with a constant step size. This step size is a function of the size of the pixel subset. If the calculated step size is by chance equal to the image width, it is increased by one, avoiding the selection of entire columns.

Table VIII contains the results with automatic selection of the colour spaces based on different subsets. Naturally, the chosen colour space differs for some images. Taking a subset of 10000 pixels pairs, in 156 of 746 cases another colour space is determined compared to the automatic selection based on all pixels. However, this marginally affects the averaged bitrate. Sometimes a colour space is selected which is even more suitable than the colour space chosen based on processing of all pixels. And often, as already mentioned above, two or more colour spaces lead to similar bitrates. Even the selection of merely 1000 pixels pairs do not significantly disturb the automatic selection of a suitable colour space. So, the computational overhead of the automatic selection is negligible.

D. Block-based selection of the colour space

In previous subsections, we have investigated colour spaces, which are optimal for the entire image. As the image content

TABLE VI
DISTRIBUTION OF BEST COLOUR SPACES $\mathbf{A}_{i,j}$, $\mathbf{B}_i$ AND RGB (PH. . . . IMAGE IS A PHOTO; O. . . . IMAGE HAS OTHER CONTENT)

		Y	G		R		B		$\frac{G+R}{2}$		$\frac{G+B}{2}$		$\frac{R+B}{2}$		$\frac{R+2G+B}{4}$		$\frac{2R+G+B}{4}$		$\frac{R+G+2B}{4}$		sum	
		i	1		2		3		4		5		6		7		8		9			
V	U	j	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.	ph.	o.
$R-G$	$B-G$	1	3	8	6	14	3	24	4	2	3	2	1	0	11	6	0	2	1	4	32	61
$R-G$	$B-(R+3G)/4$	4	8	1	11	4	14	2	22	14	5	1	0	3	32	5	8	0	3	1	103	31
$R-G$	$B-(R+G)/2$	10	15	6	13	2	44	2	54	7	7	1	5	1	84	16	7	4	17	0	246	39
$R-G$	$B-(3R+G)/4$	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
$R-G$	$B-R$	2	2	3	0	5	2	1	32	0	0	0	0	0	0	0	0	1	0	0	36	10
$B-G$	$R-(G+3B)/4$	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$B-G$	$R-(B+G)/2$	12	0	1	1	0	1	2	0	0	0	0	0	0	1	1	0	0	0	0	3	4
$B-G$	$R-(B+3G)/4$	7	2	0	2	2	3	2	4	1	6	1	1	0	21	2	4	0	1	0	44	8
$B-G$	$R-B$	3	0	2	0	6	0	2	0	0	0	1	0	0	0	0	0	0	0	0	0	11
$R-B$	$G-(R+3B)/4$	6	0	0	0	1	0	6	0	0	0	0	0	1	0	0	0	0	0	1	0	10
$R-B$	$G-(R+B)/2$	11	0	3	3	11	3	14	0	0	0	0	4	1	3	1	0	1	0	0	13	31
$R-B$	$G-(B+3R)/4$	9	0	0	3	1	0	6	0	0	0	0	1	0	2	0	3	1	0	0	9	8
		sum	32	12	40	28	71	45	115	3	21	3	12	3	154	4	22	3	23	3	488	214
see Tab. II		$\mathbf{B}_i$	1	4	2	3	3	8	0	0	1	2	0	0	2	2	0	2	0	1	9	22
		RGB																			2	11
																				total	499	247

TABLE VII
RESULTS IN BITS PER PIXEL AVERAGED OVER 746 IMAGES AND 499 PHOTOS, RESPECTIVELY, USING DIFFERENT COLOUR-SPACE SETTINGS. SEE TEXT FOR DETAILS.

colour space	averaged [bpp] all 746	photos (499)
RGB	10.214	11.769
$\mathbf{A}_{7,11}$ (YCgCo-R)	8.100	9.059
$\mathbf{A}_{3,10}$	8.030	9.005
$\mathbf{A}_{1,1}$	7.989	8.962
$\mathbf{A}_{7,1}$ (YUVr)	7.975	8.916
$\mathbf{A}_{4,10}$	7.964	8.855
$\mathbf{A}_{7,10}$	7.958	8.850
$\mathbf{A}_{7,4}$	7.953	8.857
best	7.729	8.743
automatic	7.917	8.967
automatic (incl. prediction)	7.753	8.765

TABLE VIII
RESULTS IN BITS PER PIXEL USING AUTOMATIC COLOUR-SPACE DETERMINATION ON A REDUCED SET OF PIXELS PAIRS. L IS THE NUMBER OF CASES WITH DIFFERING COLOUR-SPACE SELECTION.

number of used pixels	L	averaged [bpp] all 746	photos (499)
all	0	7.753	8.765
10000	156	7.752	8.766
5000	208	7.754	8.767
1000	402	7.760	8.770

can vary, there might be different best colour spaces dependent on the image region. For this reason, we repeat the investigations of Subsection IV-B. Now, the image is segmented in $b \times b$ blocks ($b = 2, 3, \ldots$). The block size depends on the image size.

For each block, the colour space is selected and the corresponding transform is applied. After the separate transformation of all blocks, the blocks are combined, forming three components Y, U, and V. These components are encoded in the same manner as before. There is a little overhead (one byte per block), since the numbers of all chosen colour spaces have to be sent to the decoder. The required number of bits could be reduced by conditional coding, taking into account

TABLE IX
RESULTS IN BITS PER PIXEL USING BLOCK-BASED AUTOMATIC COLOUR-SPACE DETERMINATION

number of blocks	averaged [bpp] all 746	photos (499)
1	7.753	8.765
2×2	7.746	8.758
3×3	7.742	8.751
4×4	7.741	8.749
5×5	7.742	8.749
6×6	7.742	8.748
7×7	7.743	8.748

that adjacent blocks probably have identical or similar colour spaces.

Table IX shows that the averaged bitrate decreases compared to the application of a single colour transform. However, the improvement is relatively small. The bitrates are still higher than the results of optimal colour-space selection (line 'best' in Table VII). The inspection of single results reveals that for many images the compression is worse when using the block-based application of the colour transform. In case of 3×3 segmentation, for example, one image shows a bitrate reduction by up to 0.67 bpp, while the bitrate for another image increases by more than 0.40 bpp. For 357 out of 746 images the alteration is lower than ± 0.01 bpp.

The limited improvement of the compression results is caused by the dependency between colour transform and coding algorithm. Switching from one colour space to another within an image produces not only artificial edges at the block boundaries, see **Fig. 7**, but probably also disturbs the context modelling of the LOCO-I algorithm. The application of the coding algorithm separately for each block leads to even worse results on average.

Using more, i.e. smaller, blocks improves the compression for images with distinct regional changes of the colour content. The more frequent small changes of the colours cause the alteration of the colour space, the more the coding algorithm is negatively affected. For images with homogeneous distribution of colours, the block-wise selection has little effect, as mostly

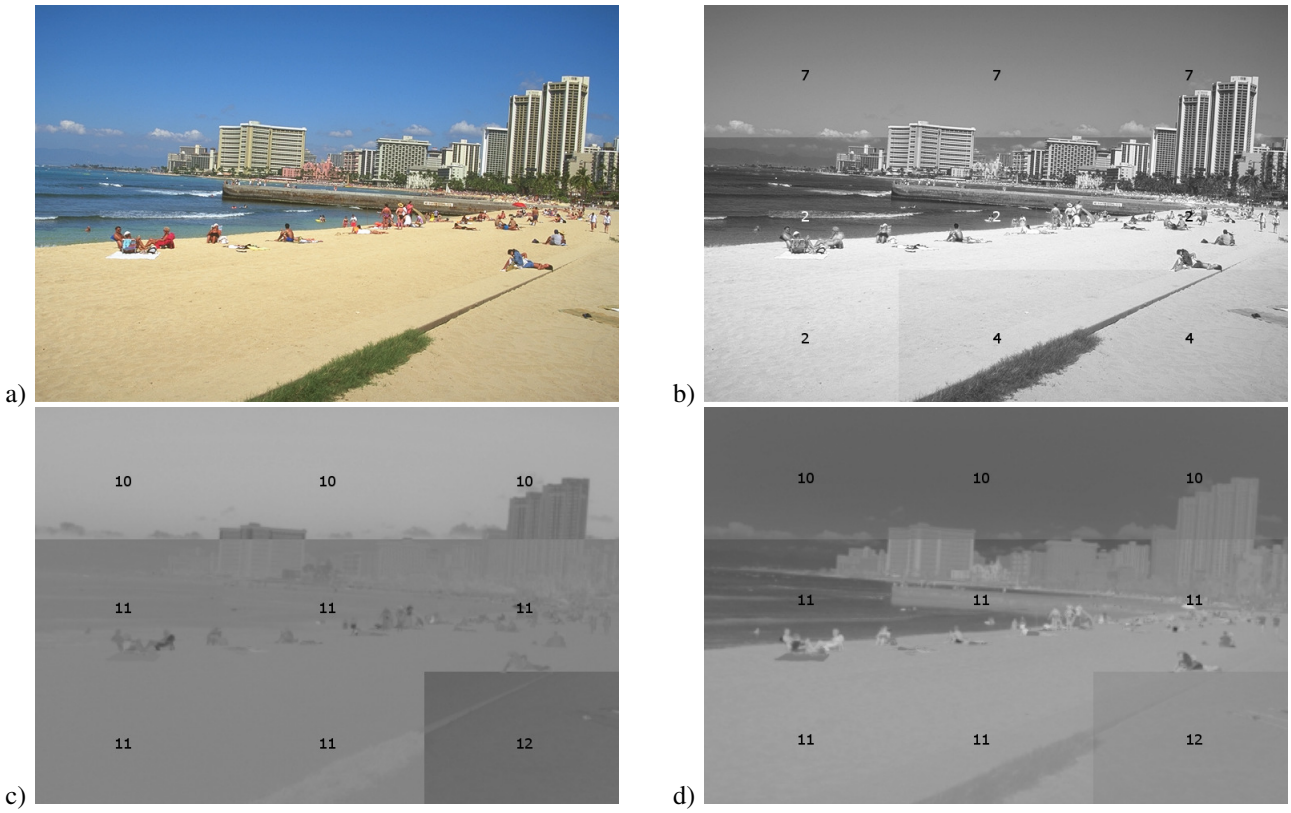

Fig. 7. Example of artificial block boundaries (edges) after 3×3 block-based colour transformation. a) original image (729×495 pixels); b) Y-component; c) U-component; d) V-component. The numbers indicate the selected computation i (in case of the Y component) and j (in case of the U/V components). The abrupt changes of the brightness at transitions from one computation to another are clearly visible. In total, there are four different colour spaces involved: $\mathbf{A}_{7,10}$, $\mathbf{A}_{2,11}$, $\mathbf{A}_{4,11}$, and $\mathbf{A}_{4,12}$. The automatically selected colour space for the entire image is $\mathbf{A}_{4,10}$.

the same colour space is chosen.

V. Evaluation of the Influence of the Compression Scheme

In previous subsections, it was already mentioned that there is a dependency between chosen colour space and performance of the coding algorithm. The question is, how compression algorithms, based on another principle than LOCO-I, can benefit from the described automatic colour-space selection.

We have modified the open source JPEG2000 codec [13] in such a manner that the standard reversible colour transformation YUVr can be substituted by the automatically selected transform. The colour space is signalled in the COD segment using those bits which are reserved for multiple component transforms.

In order to extend the variety of test images, both the LOCO-I algorithm and JPEG2000 are applied to the UCID data set, containing 1338 photographs of size 512×384 [15]. These images are holiday and sightseeing pictures as well as pictures of various colourful objects.

Table X shows how often a particular colour space leads to the lowest bitrate in comparison to the automatically selected colour space.

The assignment of rows and columns with respect to i and j is identical to Table VI. The numbers allow the following interpretation:

- The most effective colour spaces when using LOCO-I are $\mathbf{A}_{7,7}$ ($208\times$) and $\mathbf{A}_{5,7}$ ($140\times$) followed by $\mathbf{A}_{7,11}$ ($124\times$).
- The most effective colour spaces when using JPEG2000 are $\mathbf{A}_{7,11}$ ($197\times$) and $\mathbf{A}_{7,7}$ ($170\times$) followed by $\mathbf{A}_{5,1}$ ($112\times$) .
- YCgCo-R ($\mathbf{A}_{7,11}$) leads more often to the smallest bitrate than YUVr ($\mathbf{A}_{7,1}$) for both, LOCO-I and JPEG2000.
- Colour spaces using the computation of U and V based on $j = 2, 3, 5, 8$ never lead to the lowest bitrate.
- Colour spaces using the computation of Y based on $i = 6$ never lead to the lowest bitrate.
- The distribution is distinctly more pronounced than the distribution for the set of images with varying sources including synthetic images (Tab. VI).

There is also a high correlation between the automatic selection and the best colour spaces for LOCO-I and JPEG2000. The linear correlation coefficients

$$\rho(X, Y) = \frac{\mathrm{cov}(X, Y)}{\sigma_X \cdot \sigma_Y} \tag{20}$$

with X and Y being the counts of each colour space are

X	Y	$\rho(X,Y)$
automatic	LOCO-I	0.864
automatic	JPEG2000	0.949
LOCO-I	JPEG2000	0.946

The measure takes only the distribution of the colour spaces into account and tells nothing about the correct selection of

TABLE X
DISTRIBUTION OF BEST COLOUR SPACES $\mathbf{A}_{i,j}$, $\mathbf{B}_i$ AND RGB FOR THE UCID DATA SET COMPARED TO THE AUTOMATIC SELECTION (LO ... LOCO-I COMPRESSION; J2K ... JPEG2000 COMPRESSION; AUT. ... AUTOMATIC SELECTION)

i	1			2			3			4			5			6			7			8			9		
j	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.	Lo	J2K	aut.
1	26	39	49	4	4	4	23	20	35	10	9	11	75	112	146	0	0	1	58	80	114	0	2	1	1	2	5
4	14	16	29	1	1	4	1	2	2	16	9	6	29	27	25	0	0	0	111	78	51	0	0	1	1	0	1
10	3	4	1	1	1	0	0	1	0	0	0	0	0	0	2	0	0	0	4	5	2	0	0	0	1	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	16	13	13	1	1	0	1	0	0	1	1	1	49	31	19	0	0	0	109	78	54	0	0	0	0	0	0
7	63	77	48	2	2	4	8	12	8	10	11	5	140	115	91	0	0	0	208	170	142	0	0	1	2	0	2
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	7	3	5	1	2	3	5	5	23	0	0	3	49	54	43	0	0	0	57	47	44	0	0	0	0	0	3
11	18	12	9	2	1	2	6	4	16	8	9	23	43	54	103	0	0	1	124	197	158	1	1	1	0	1	3
9	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	0	5	0	0	0	0	0	0
sum	147	164	154	12	12	19	44	44	84	45	40	50	385	393	430	0	0	2	671	655	570	1	3	4	5	3	14
$\mathbf{B}_i$	2	3	5	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	3	1	5	3	1	0	0	0
RGB																									16	11	4
																						total			1338	1338	1338

the colour spaces. Still, we see that the automatic selection is surprisingly more correlated with the results for JPEG2000 than with the results of LOCO-I. The direct comparison of automatically chosen and best colour spaces reveals that the automatic selection leads in 537 cases to the best colour space when using LOCO-I and in 696 cases to the best for JPEG2000. In order to find an explanation for this effect, the left-neighbour prediction in the colour-space-selection process is substituted by the median-edge-detection predictor (MED), which is also used in LOCO-I. Interestingly, the automatic selection does not only improve for the LOCO-I compressor, but also for JPEG2000. The correlation coefficients increase to

X	Y	$\rho(X,Y)$
automatic (MED)	LOCO-I	0.962
automatic (MED)	JPEG2000	0.973

and the best colour space is now selected in 724 cases (LOCO-I) and 812 cases (JPEG2000), respectively.

Table XI allows a comparison of the compression results. Though YCgCo-R is more often used as best colour space, its average performance is worse than the performance of YUVr for both compression algorithms. The results also show that it is indeed reasonable to choose the colour space adaptively. While for the data set used in Section IV $\mathbf{A}_{7,10}$ is the best choice for photos, it is now the $\mathbf{A}_{7,7}$ colour space and $\mathbf{A}_{7,10}$ is even worse than $\mathbf{A}_{1,1}$. There is a strong suspicion that these differences are stemming from the type of the camera (three-chip or single-chip), and the best colour space also depends on how the RGB data are computed from the Bayer pattern in single-chip cameras. However, the clarification of this assumption requires more investigations.

Using the MED predictor improves the automatic colour-space selection and consequently the averaged bitrates decrease. The additional computation costs can be compensated by reducing the number of pixels, which are included in the determination of the selection process. The results in Table XI show that using the MED prediction on a set of 5000 pixel quadruples[2] is still better (12.023 / 12.426 bpp) than using the left-neighbour prediction on the full set (12.031 / 12.431 bpp).

TABLE XI
RESULTS IN BITS PER PIXEL AVERAGED OVER 1338 IMAGES FROM THE UCID DATA SET (SEE TEXT FOR DETAILS)

colour space	averaged [bpp] LOCO-I	JPEG2000
RGB	13.337	13.922
$\mathbf{A}_{7,10}$	12.189	12.604
$\mathbf{A}_{1,1}$	12.141	12.530
YCgCo-R	12.116	12.514
$\mathbf{A}_{7,4}$	12.093	12.500
YUVr	12.088	12.485
$\mathbf{A}_{7,7}$	12.066	12.471
best	12.013	12.418
automatic (incl. prediction)	12.031	12.431
automatic (incl. MED)	12.021	12.424
automatic (incl. MED, 5000)	12.023	12.426
automatic (incl. MED, 1000)	12.032	12.436
automatic, MED, 2x2 blocks	12.010	12.419
automatic, MED, 3x3 blocks	12.002	12.416
automatic, MED, 4x4 blocks	12.001	12.422
automatic, MED, 5x5 blocks	12.001	12.428
aut., MED, 3x3 blocks, 10000	12.010	12.427
aut., MED, 3x3 blocks, 5000	12.017	12.435

The increased level of adaptivity with respect to the automatic selection of the colour space by using 2×2 or more blocks again improves the compression. The bit savings are lower for JPEG2000, probably because the wavelet transform less successfully adapts to the artificial edges, which can appear if neighbouring blocks are converted with different colour transforms, see Subsection IV-D.

Finally, we have investigated the block-based adaptation in combination with the MED predictor and reduced sets of pixels. When only 10000 pixel quadruples are investigated (with respect to entire image) and the 3×3 segmentation is applied, the compression result for LOCO-I (12.010 bpp) is better than using a single colour space and automatic selection based on the full set of pixels (12.021 bpp). The

[2]In contrast to the left-neighbour prediction, the MED predictor requires three neighbours for the determination of the estimation value.

same comparison, but using JPEG2000, shows almost identical averaged bitrates (12.427 bpp and 12.424 bpp).

As some colour spaces are seldom selected in the case of the UCID data set, the question is, whether the set of colour transforms can be reduced in order to further lower the computational costs. Therefore, we have excluded the Y components $i = 2, 6, 8, 9$ and the UV combinations $j = 2, 3, 5, 8, 9, 10$ as well as all colour spaces from the $\mathbf{B}_i$ series. Using LOCO-I and the automatic selection including the MED predictor, this concerns in total 36 images, and the average bitrate increased from 12.021 bpp to 12.023 bpp. Enabling in addition the 3×3 segmentation and limiting the investigated pixels to 10000 quadruples, 427 images are affected and the bitrate increases slightly from 12.010 bpp to 12.014 bpp. However, it must be pointed out again that the exclusion of colour transforms can have severe effects when the source of the images is unknown or can vary.

VI. Summary and discussion

We have presented a new family of 108 multiplierless reversible colour transformations based on a unique processing structure enabling a broad range of possible computations of the luminance and the chrominances. The set of colour transforms has been supplemented with nine colour spaces with a single difference (chrominance) component and two other components.

Based on a broad variety of natural images (photos), computer-generated images, and images with mixed content, it could be shown that the adaptive selection of the colour space significantly improves the compression performance. Furthermore, it could be proven that this selection can be done automatically with negligible loss of performance compared to the optimal selection.

The proposed approach to automated selection incorporates spatial decorrelation. This is essential, as the colour transformation can disturb the spatial dependencies between signal values if noise from one colour component is spread over the other components. The subsequent spatial decorrelation step would result in a signal with higher entropy, leading to worse compression. The used prediction step takes these dependencies into account. So, the colour space is not chosen according to the best colour decorrelation but according to the presumably minimum bitrate. The results show that this approach is successful, not only for prediction-based compression, but also for wavelet-based compression schemes.

Even though the investigated colour transform can be performed with minimum computational efforts, the family samples the infinite range of possible transforms finely enough. The variety is limited to a reasonable subset, with typically two or more colour spaces leading to bitrates being equal or close to the lowest one. If the source of images is known and fixed, it is possible to reduce the variety of different colour transforms without significant loss in compression performance.

The complexity of the automatic colour-space selection is relatively low. First, the different colour spaces always share one or two of the computations for Y, U and V. Each variant has to be computed only once. Secondly, the selection of Y and the selection of the U-V combination can be done separately. This reduces the number of comparisons. Thirdly, it is not necessary to access all pixels of the image to be transformed. A subset of, for example, 10000 pixel quadruples (in the case of MED prediction) selected from all regions of the image are sufficient for a colour-space selection close to the optimum.

The adaptation of the colour transform can be increased by a block-based selection. However, the switching between different colour spaces introduces non-linearity, which adversely affects the compression. That is why the block-based selection leads to only little improvement on average.

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

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.S.; Sullivan, G.J.; Srinivasan, S.: Lifting-Based Reversible Color Transformations for Image Compression. *Proc. of SPIE* , Vol.7073, 11 August 2008, San Diego, CA, USA

[4] 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

[5] Strutz, T.: Adaptive Selection of Colour Transformations for Reversible Image Compression. *Proc. of EUSIPCO 2012*, 27-31 August, 2012 Bucharest, Romania, 1204–1208

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

[7] 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

[8] 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

[9] 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

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

[11] Matsuda, I.; Kaneko, T.; Minezawa, A.; Itoh, S.: Lossless Coding of Color Images using Block-Adaptive Inter-Color Prediction . *Proc. of ICIP 2007*, Sep. 16-19, 2007, 329 – 332

[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

[13] http://www.openjpeg.org last visited: 16.04.2012

[14] http://www1.hft-leipzig.de/strutz/Papers/Testimages/CT2/ last visited: 24.10.2012

[15] Schaefer, G.; Stich, M.: UCID - An Uncompressed Colour Image Database. *Proc. of SPIE*, Vol. 5307, 2004, 472–480

Tilo Strutz holds a Dipl.-Ing. degree in Electrical Engineering (1994), a Dr.-Ing. degree in Signal Processing (1997), and a Dr.-Ing. habil. degree in Communications Engineering (2002) from the University of Rostock, Germany. He worked at the European Molecular Biology Laboratory (Outstation Hamburg) in the field of multidimensional signal processing and data analysis from 2003 to 2007. Dr. Strutz is now professor of information and coding theory at the Leipzig University of Telecommunications (Hochschule für Telekommunikation Leipzig). His research interests range from general signal processing to special problems of image processing, to data compression.