This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

Reversible Colour Spaces without Increased Bit Depth and Their Adaptive Selection

Tilo Strutz and Alexander Leipnitz

Abstract

The efficient compression of colour images requires a processing step exploiting the correlation between the colour components. This is typically realised using a colour transformation. In lossless compression systems, the reversible colour transformation increases the bit-depth for chrominance components from eight to nine bits per pixels. This can be avoided by using modulo arithmetic, while keeping the property of reversibility. This paper investigates the impact of these modulo operations on the compression performance, compares different processing structures, and proposes a new adaptive selection of suitable colour spaces. It is shown that (i) the limitation of the bit depth generally leads to lower compression performance, (ii) the drop in performance depends on the processing structure used, and (iii) the average performance can be improved by the proposed adaptive selection of the colour space.

Index Terms

reversible colour transformation, lossless image compression, colour-space selection

Tilo Strutz and Alexander Leipnitz are with the Deutsche Telekom, Leipzig University of Telecommunications, Institute of Communications Engineering, Gustav-Freytag-Str. 43–45, 04277 Leipzig, Germany

I. INTRODUCTION

The performance of image-compression systems depends on, aside from other aspects, the exploitation of correlations between the colour components. Investigations have shown, that the combined processing of information stemming from different colour components is superior to the separate treatment of these components [1], [2]. Unfortunately, this is accompanied by a distinct increase in computational complexity. Therefore, the standard approach today is to sequence the processing in terms of applying a separate colour transformation before the colour components are independently treated. In [3] it had been shown that the colour decorrelation can be optimised with respect to the coding gain using an adaptive selection of one out of a large variety of colour spaces.

When applied to lossless image compression, the colour transformation has to be reversible. Reversible colour transformations (RCT) can be easily implemented using lifting structures and rounding of intermediate results to integer values [4]. In general, the RGB (red, green, blue) input is converted to a YUV output. While the luminance component Y typically keeps its bit depth, the chrominances U and V require one bit more after the transformation. Depending on the coding method, these additional bits can complicate the algorithm or, if implementation in hardware is addressed, the chip costs will be higher. Some methods even cannot process components with higher bit depth. We will discuss only the common case of 24-bit RGB data in the following.

In colour transformations, the increased range of values is caused by summation operations. Using modulo arithmetic, these values can be mapped back into the original range, while keeping the property of reversibility. In JPEG-LS Part-2, for example, this is the standard procedure [5]. This is one major reason to extend the work of [3] towards the investigation of RCTs without increased bit depth in a comprehensive manner. In the sequel, we call transformations mapping

24-bit RGB data to 24-bit YUV data '24-bit RCT'. Investigations are already taking place into the performance of some 24-bit RCTs [6].

Using colour spaces with such modulo operations, the question remains whether the colour decorrelation can somehow be improved. This paper investigates the influence of modulo operations on compression performance and proposes a method for adaptive selection of the most suitable colour space.

The paper is organised as follows: Section II describes the processing of RCTs with modulo operations. Section III compares conventional colour transformations and 24-bit RCTs and investigates the impact of using modulo operations. Section IV reviews the adaptive selection proposed in [3], introduces required modifications, and presents results. A summary is given in Section V.

II. Reversible transformations with modulo operations

The limitation of the precision of samples to eight bits demands a modulo operation after those steps, which could result in values outside the maximum range. With respect to the signal flow depicted in **Figure 1**, we must insert this operation after each summation. With $\varepsilon = 0$ and $\alpha_1 = \alpha_2 = 0.25$ (and without modulo operations) this structure corresponds to $\mathrm{YUV_r}$, the RCT defined for JPEG 2000 [7]: $V = R - G$, $U = B - G$, $Y = G - \lfloor (V + U)/4 \rfloor$.

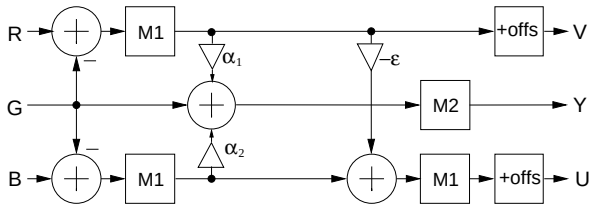
There are four modulo operations (M1, M2) in the forward transformation and also four in the backward transformation. A modulo operation is defined as

$$x' = \begin{cases} x - r & \text{if} \quad x > u \\ x + r & \text{if} \quad x < l \\ x & \text{else} \end{cases} \tag{1}$$

The interval limits used are $l = -128$ and $u = 127$ for M1 and $l = 0$ and $u = 255$ for M2. The interval range is $r = 256$ in both cases. Obviously, one modulo operation in Figure 1 can be

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

a)



b)

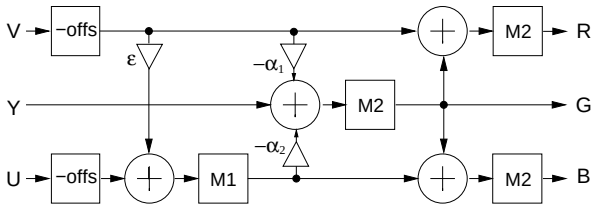


Fig. 1. Flow chart of the reversible 24-bit colour transformation of a single RGB triple and its inverse, without depiction of the rounding operations

skipped if $\varepsilon = 0$; the same holds true for the condition $\alpha_1 = \alpha_2 = 0$. The operation called '+offs' simply adds a value of 128 if non-negative values are expected in the subsequent processing stage. This offset must be removed before the data are transformed back to the RGB domain (Fig. 1b). Depending on the coefficients α_1, α_2, ε and a possible permutation of the RGB input values, $9 \times 12 = 108$ different colour spaces are allocatable (**Tab. I**). These colour spaces are called $A_{i,j}$ and can be realised without any multiplication.

In addition, we include the transformation

$$\begin{aligned} V &= R - G \\ U &= B - \lfloor (87 \cdot R + 169 \cdot G)/256 \rfloor \\ Y &= G + \lfloor (86 \cdot V + 29 \cdot U)/256 \rfloor \end{aligned} \tag{2}$$

proposed in [8] in our investigations, as it shows good decorrelation for many images. It will be called 'Pei09' in the sequel. With respect to the structure in Figure 1, this colour space can be implemented with $\alpha_1 = 76/256$, $\alpha_2 = 29/256$, and $\varepsilon = -87/256$ saving one multiplication

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

TABLE I

Y, U, AND V COMPONENTS OF COLOUR SPACES $A_{i,j}$

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

j	V	U
1	$R-G$	$B-G$
2	$G-R$	$B-R$
3	$R-B$	$G-B$
4	$R-G$	$B-\lfloor (R+3G)/4 \rfloor$
5	$G-R$	$B-\lfloor (G+3R)/4 \rfloor$
6	$R-B$	$G-\lfloor (R+3B)/4 \rfloor$
7	$B-G$	$R-\lfloor (B+3G)/4 \rfloor$
8	$G-B$	$R-\lfloor (G+3B)/4 \rfloor$
9	$B-R$	$G-\lfloor (B+3R)/4 \rfloor$
10	$R-G$	$B-\lfloor (R+G)/2 \rfloor$
11	$R-B$	$G-\lfloor (R+B)/2 \rfloor$
12	$B-G$	$R-\lfloor (B+G)/2 \rfloor$

while increasing the number of summations.

III. COMPARISON OF CONVENTIONAL RCTS AND 24-BIT RCTS

The big disadvantage of using modulo operations is that it leads to many discontinuities in the transformed signal, as can be seen in **Figure 2**. This can adversely affect the subsequent processing, especially the prediction of signal values. In the following, we investigate the influence of the usage of modulo operations on the compression performance.

A. Colour transformation and LOCO-I compression

Table II shows the compression performance in bits per pixel if fixed colour transformations are applied and the resulting colour components are separately compressed using the LOCO-I

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

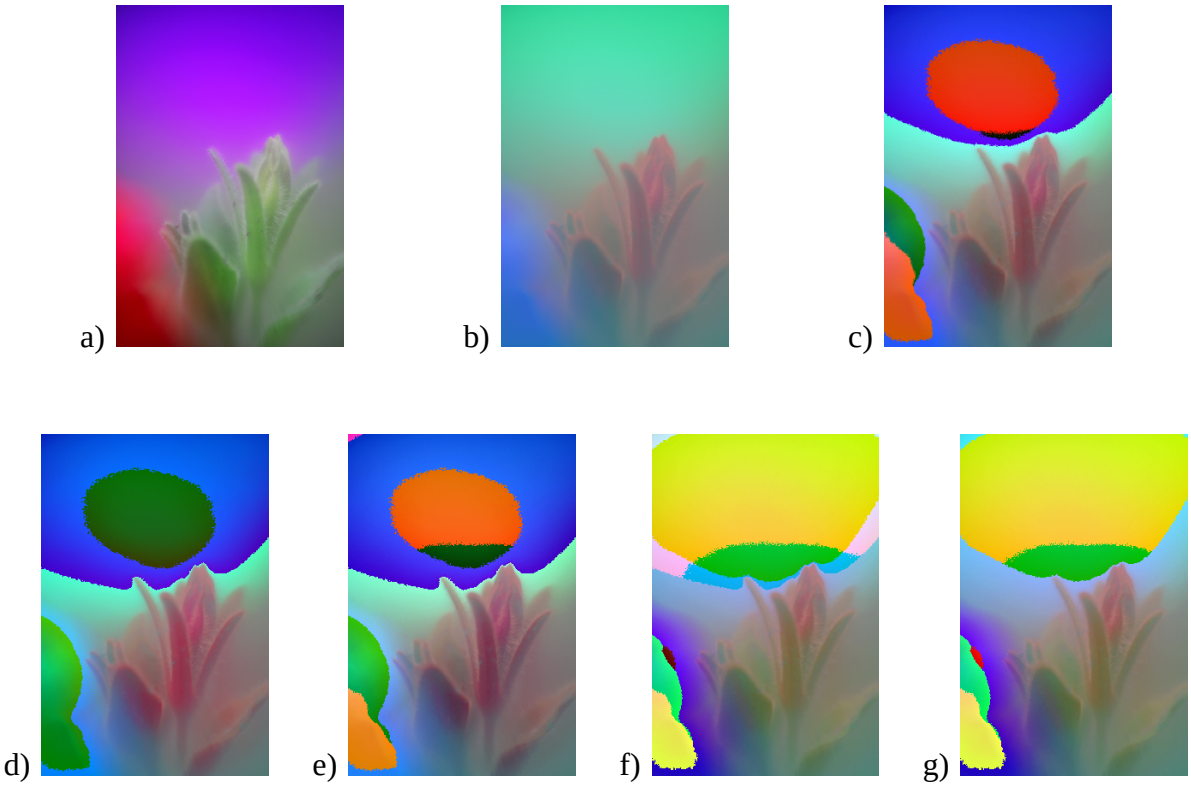


Fig. 2. Influence of modulo operations for different colour spaces, a) original RGB image, b) YCbCr colour space ([9], lossy), c) colour space according to equation (2), d) $A_{1,1}$ colour space, e), $A_{7,1}$ (YUV$_r$) f) $A_{7,11}$ (YCgCo-R) g) $A_{7,11}$ (using the structure in Fig.3); c) – g) including modulo operations; images b) - g) are shown without inverse colour transformation

[10] algorithm. The results are averaged over 746 images (499 photographs and 247 computer-generated images) taken from [11]. The differences show that the compression performance drops on average when the modulo arithmetic is applied (24-bit RCT). Only for colour spaces where the Y component is simply a copy of R, G, or B (e.g., $A_{1,1}$, $A_{2,1}$, and $A_{3,1}$) are the results almost the same. There are fewer modulo steps here and consequently a lesser chance that a mapping of signal values will occur. Obviously, the median-edge-detection (MED) predictor of LOCO-I can cope with the unavoidable artificial edges in the chrominance components. In [6] even an improved performance for some special 24-bit RCTs ($A_{1,1}$, $A_{2,1}$) was reported in application to specific sets of images. This could be verified when using the JPEG-LS compressor from [12]

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

TABLE II

COMPARISON OF COMPRESSION PERFORMANCE IN BIT PER PIXEL DEPENDENT ON WHETHER THE MODULO OPERATIONS ARE USED OR NOT; USING LOCO-I COMPRESSION AVERAGED OVER 746 IMAGES

colour space	RCT	24-bit RCT	diff.
RGB	10.214	10.214	0.000
$A_{1,1}$	7.989	**7.990**	-0.001
$A_{1,4}$	7.967	7.995	-0.028
$A_{2,1}$	8.013	8.015	-0.002
$A_{3,1}$	8.047	8.048	-0.001
$A_{3,10}$	8.030	8.071	-0.041
$A_{4,10}$	7.964	8.041	-0.077
$A_{7,1}$ (YUVr)	7.975	8.024	-0.049
$A_{7,4}$	**7.953**	8.028	-0.075
$A_{7,10}$	7.958	8.047	-0.089
$A_{7,11}$ (YCgCo-R)	8.099	8.306	-0.207
eq. (2), Pei09	7.955	8.003	-0.048
best	7.726	7.749	-0.023

but not when using the LOCO-I implementation. In general, the influence of modulo operations also depends on the image content.

If one selects for each image the colour space resulting in the lowest bitrate (line ‘best’ in Table II), then the usage of modulo operations drops the average compression by -0.023 bits per pixel.

B. Colour transformation and CoBaLP2 compression

The previous subsection showed that the performance of the LOCO-I algorithm is almost unaffected by the modulo operation for some colour spaces. The question is, whether this observation can also be generalised for other compression systems. For this, the investigations have

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

TABLE III

COMPARISON OF COMPRESSION PERFORMANCE IN BIT PER PIXEL WHEN COBALP2 IS USED, AVERAGED OVER 746 IMAGES

colour space	RCT	24-bit RCT	diff.
RGB	9.537	9.537	0.000
$A_{1,1}$	7.403	**7.480**	-0.077
$A_{1,4}$	7.400	7.507	-0.107
$A_{2,1}$	7.436	7.513	-0.077
$A_{3,1}$	7.483	7.561	-0.078
$A_{3,10}$	7.474	7.597	-0.123
$A_{4,10}$	**7.393**	7.570	-0.177
$A_{7,1}$ (YUVr)	7.402	7.560	-0.158
$A_{7,4}$	7.399	7.586	-0.187
$A_{7,10}$	**7.393**	7.596	-0.203
$A_{7,11}$ (YCgCo-R)	7.509	7.856	-0.347
Pei09	7.409	7.572	-0.163

been repeated using the compression system proposed in [13], which uses a more sophisticated context-based linear prediction (CoBaLP2) technique in combination with arithmetic coding.[1] It can clearly be seen in **Table III** that the performance of this compression system is much higher compared to LOCO-I. It is worth mentioning that the best results on average are achieved with the colour spaces $A_{4,10}$ and $A_{7,10}$ (column 'RCT') and not with $A_{7,4}$ as in combination with LOCO-I. The influence of the modulo operation is generally greater when using COBaLP2 and a distinct loss in performance can be observed for all colour spaces.

[1]The template matching was switched off speeding up the investigations.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

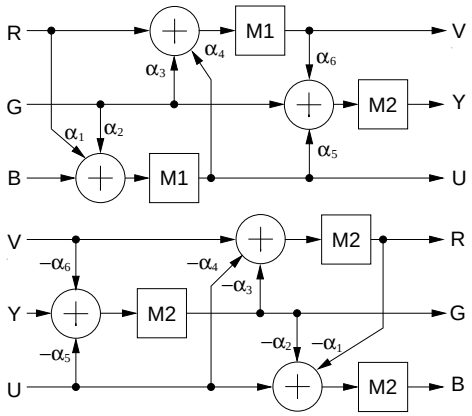


Fig. 3. Flow chart of 24-bit RCTs according to JPEG-LS standard, without depiction of the rounding operations

C. Structure with reduced number of modulo operations

The JPEG-LS standard [5] defines colour transformations, which can be described with the processing structure in **Figure 3**. Compared to Figure 1, the number of modulo operations is reduced from four to three ('3-sum structure'). However, the realisation of many of the colour transformations proposed in [3] now require at least one multiplication. The conversion from RGB to YUV is given in Figure 3 by

$$\begin{aligned} U =& \alpha_1 \cdot R + \alpha_2 \cdot G + B \\ V =& (1 + \alpha_1\alpha_4) \cdot R + (\alpha_3 + \alpha_2\alpha_4) \cdot G + \alpha_4 \cdot B \\ Y =& (\alpha_1\alpha_5 + \alpha_1\alpha_4\alpha_6 + \alpha_6) \cdot R \\ & + (1 + \alpha_2\alpha_5 + \alpha_2\alpha_4\alpha_6 + \alpha_3\alpha_6) \cdot G \\ & + (\alpha_5 + \alpha_4\alpha_6) \cdot B \quad . \end{aligned} \tag{3}$$

For example, the colour space $A_{7,10}$ can be realised using the coefficients $\alpha_1 = \alpha_2 = -1/2$, $\alpha_3 = -1$, $\alpha_4 = 0$, $\alpha_5 = 1/4$, and $\alpha_6 = 3/8$. Obviously, α_6 requires a multiplication with three or, alternatively, two additional summations ($3x = x + x + x$). The colour space $A_{7,11}$ is realised

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

TABLE IV

COMPARISON OF COMPRESSION PERFORMANCE IN BIT PER PIXEL IN DEPENDENCE ON THE RCT'S PROCESSING STRUCTURE USING LOCO-I, AVERAGED OVER 746 IMAGES

	Structure according to		
colour space	Fig. 1	Fig. 3	diff.
$A_{1,4}$	7.995	7.968	-0.027
$A_{4,4}$	8.022	7.996	-0.028
$A_{5,4}$	8.046	8.040	-0.006
$A_{6,4}$	8.102	8.076	-0.026
$A_{7,4}$	8.028	8.008	-0.020
$A_{8,4}$	8.055	8.029	-0.026
$A_{9,4}$	8.073	8.061	-0.012
$A_{1,10}$	8.013	7.975	-0.038
$A_{4,10}$	8.041	8.002	-0.039
$A_{5,10}$	8.065	8.071	0.006
$A_{6,10}$	8.120	8.089	-0.031
$A_{7,10}$	8.047	8.027	-0.020
$A_{8,10}$	8.074	8.043	-0.031
$A_{9,10}$	8.102	8.086	-0.016

with $\alpha_1 = \alpha_3 = -1$, $\alpha_2 = \alpha_5 = 0$, $\alpha_4 = \alpha_6 = 1/2$.

The assumption that a reduced number of modulo operations per pixel benefits the compression could be verified. **Table IV** compares the average results for selected colour transformations, which use $\varepsilon \neq 0$ in the structure of Figure 1.

The result of $A_{1,4}$ using the 3-sum structure is even about the same as the result without modulo operations. Interestingly, the 3-sum structure is not always better than the structure according to Figure 1 (see $A_{5,10}$). Upon closer inspection, it can be observed that the advantage of one structure over the other depends on the content of each individual image.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

IV. Adaptive Selection of colour space

In some applications, there might be rigid reasons to use a 24-bit RCT. Then, the algorithm should make the best of it and should adaptively select a suitable colour space instead of using a fixed one. In [3] the automatic choice of a colour space was based on an entropy criterion. It had been proposed to firstly compute all possible variants of chrominance (U, V) and luminance (Y) signals. Since the colour spaces share the same components in different combinations, the computational overhead is much smaller than the number of colour spaces would indicate. Secondly a simple prediction (MED) has to be applied to each component leading to prediction error signals E_{Y}, E_{U}, and E_{V}. Thirdly, the entropies H must be computed separately for these prediction-error components. Finally, the colour space with the smallest sum entropy

$$H(E)_{\mathrm{sum}} = H(E_{\mathrm{Y}}) + H(E_{\mathrm{U}}) + H(E_{\mathrm{V}}) \tag{4}$$

is selected as most suitable colour space. Furthermore, it had been proven that a subset of all pixels is enough for this process, which enormously reduces the computational costs of the colour-space selection.

For the sake of completeness, we also include RGB and the B_i colour spaces, which are proposed for weakly correlated data in [3] (**Fig. 4** and **Tab. V**). As can be seen, they require less modulo operations.

The investigation with 24-bit RCTs (and processing structure according to Fig. 1) shows that the approach described in [3] is not adequate in the presence of modulo operation (**Tab. VI**, comparison of rows ‘best’ and ‘automatic [3]’). The adaptive selection leads to better results on average than any fixed colour space (compare Tab. II), however, there is too great a gap between this and the optimum (line ‘best’). To overcome this problem, the modulo operations also have to be applied in the process of colour-space selection. As a result, the averaged bitrate is much closer to the theoretical optimum (Tab. VI, lines ‘automatic (proposed)’ and ‘best’).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LSP.2015.2397034, IEEE Signal Processing Letters

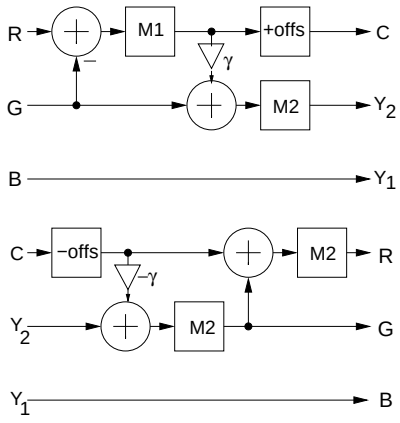


Fig. 4. Flow chart of 24-bit RCTs according to colour spaces B_i

TABLE V

COMPONENTS OF COLOUR SPACES B_i

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

V. SUMMARY AND DISCUSSION

We have investigated the impact of modulo operations in 24-bit RCTs and found that they generally decrease the compression performance on average. In particular, more advanced prediction schemes (e.g., CoBaLP2) are sensitive to the discontinuities caused by the modulo operations.

Furthermore, we were able to show that the performance depends on the structure for pro-

TABLE VI

COMPRESSION PERFORMANCE IN BIT PER PIXEL WITH ADAPTIVE SELECTION OF 24-BIT COLOUR SPACES, AVERAGED OVER 746 IMAGES

colour space	LOCO-I	CoBaLP2
best	7.749	7.239
automatic [3]	7.809	7.344
automatic (proposed)	7.776	7.282

cessing the colour transformation. A structure similar to the definition in the standard JPEG-LS (Part 2) yields somewhat better results when the number of modulo operations can be reduced. Unfortunately, the complexity of some of the best colour transformations is increased using this structure.

The adaptive selection of 24-bit RCTs is possible, similar to the approach in [3]. However, the modulo operations also have to be taken into account in the selection process in order to obtain a sufficient performance.

The programs and images are available at [11].

REFERENCES

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

[2] Pasteau, F.; Strauss, C.; Babel, M.; Déforges, O.; Bédat, L.: Improved colour decorrelation for lossless colour image compression using the LAR codec. *EUSIPCO 2009*, Glasgow, Scotland, 24-28 August 2009, 2122–2126

[3] Strutz, T.: Multiplierless Reversible Colour Transforms and their Automatic Selection for Image Data Compression. *IEEE Transactions on Circuits and Systems for Video Technology*, Vol.23, No.7, July 2013, 1249-1259

[4] 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

[5] ISO/IEC 14495-2, *Information technology – Lossless and near-lossless compression of continuous-tone still images: Extensions*. International Standard, second edition, 1 April 2003

[6] Starosolski, R.: New simple and efficient color space transformations for lossless image compression. *J. Visual Communication and Image Representation*, Vol.25, No.5, 2014, 1056–1063

[7] ISO/IEC FCD 15444-1, *Information technology – JPEG 2000 Image Coding System*. JPEG 2000 Final Committee Draft Version 1.0, 16 March 2000

[8] Pei, S.-Ch.; Ding, J.-J: Improved reversible integer-to-integer color transforms. *Proc. of IEEE ICIP 2009*, Cairo, Egypt, 7-10 Nov. 2009, 473 – 476

[9] Hamilton, E.: *JPEG File Interchange Format*, Version 1.02, September 1, 1992, http://www.w3.org/Graphics/JPEG/jfif3.pdf, last visted on 20 November 2014

[10] Weinberger, M.J.; Seroussi, G.; Sapiro, G.: LOCO-I: A Low Complexity, Context Based, Lossless Image Compression Algorithm. *Proc. of Data Compression Conference, DCC'96*, March 1996, Snowbird, Utah, USA, 1996, 140–149

[11] http://www1.hft-leipzig.de/strutz/Papers/24bitRCT_resources/ last visited 12 January 2015

[12] Jakulin, A: JPEG-LS Public Domain Code. http://www.stat.columbia. edu/∼jakulin/jpeg-ls/jpeg_ls_v2.2.tar.gz last visited 10 January 2015

[13] Strutz, T.: Adaptive context formation for linear prediction of image data. *IEEE ICIP 2014*, Paris, France, 27-30 Oct. 2014, 5631 – 5635