J. Vis. Commun. Image R. 25 (2014) 1056–1063

Contents lists available at ScienceDirect

J. Vis. Commun. Image R.

journal homepage: www.elsevier.com/locate/jvci

New simple and efficient color space transformations for lossless image compression

Roman Starosolski

Institute of Computer Science, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland

ARTICLE INFO

Article history:
Received 17 December 2013
Accepted 5 March 2014
Available online 15 March 2014

Keywords:
Reversible color space transformation
Lossless image compression
Human vision system
Image coding
RCT
YCoCg-R
Karhunen–Loève transformation
JPEG-LS
JPEG2000
JPEG XR

ABSTRACT

We present simple color space transformations for lossless image compression and compare them with established transformations including RCT, YCoCg-R and with the optimal KLT for 3 sets of test images and for significantly different compression algorithms: JPEG-LS, JPEG2000 and JPEG XR. One of the transformations, RDgDb, which requires just 2 integer subtractions per image pixel, on average results in the best ratios for JPEG2000 and JPEG XR, while for a specific set or in case of JPEG-LS its compression ratios are either the best or within 0.1 bpp from the best. The overall best ratios were obtained with JPEG-LS and the modular-arithmetic variant of RDgDb (mRDgDb). Another transformation (LDgEb), based on analog transformations in human vision system, is with respect to complexity and average ratios better than RCT and YCoCg-R, although worse than RDgDb; for one of the sets it obtains the best ratios.

1. Introduction

It is known, that red, green, and blue primary color components of the RGB color space are highly correlated for natural images. The high correlation indicates that more than one image component contains the same information, e.g., image area which is bright in green component usually is also bright in red and blue. Above usually is true also for computer generated images since artificial images mostly are made to resemble natural ones, however it depends on the actual objective of the image's creator. The most common approach to RGB color image compression is to compress independently the image components obtained using a transformation from RGB to some less correlated color space. Without the transformation we would unnecessarily compress the same information more than once.

For a specific image, using on it the Principal Component Analysis (PCA) we may obtain the image-dependent Karhunen–Loève transformation (KLT), which optimally decorrelates the image [1]. Since PCA/KLT is practically too time complex to be computed each time an image gets compressed, fixed transformations are constructed by performing PCA on a representative set of images. Then it is assumed that the obtained KLT transformation will match individual images from and outside of the used set. However note, that optimal decorrelation of color space of the set of images may not lead to the best compression ratios of individual images — since, among other things, actual inter-component dependencies may be different in various images or even in various regions of the same image; also the transformation while removing inter-component correlation may transfer incompressible noise from one component to another. Many transformations were constructed based on KLT; recently different approaches allowing adaptation of the color space transformation to a given image were proposed. In [2] an adaptive selection of transformation, from a large family of simple transformations, is done at the cost of slight increase of the color image transformation process complexity. Significantly more complex, yet simpler than computing PCA/KLT for the whole image, Singular Value Decomposition based, image adaptive method of constructing color space transformation for the lossy compression is presented in [3]. Decades ago a PCA/KLT transformation constructed for video data with additional requirement to obtain one component that approximates the intensity perception of the human vision system, was used to construct the YCbCr color space [4]. The YCbCr color space is up to today used in various television systems and in lossy compression algorithms. Several variants of the space and of transformations from RGB to YCbCr exist. One of them (ICT), used in JPEG2000 [5] for lossy compression, is presented below with its inverse (Eq. (1)).

E-mail addresses: rstarosolski@polsl.pl, rstaros@gmail.com

http://dx.doi.org/10.1016/j.jvcir.2014.03.003
1047-3203/

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.16875 & -0.33126 & 0.5 \\ 0.5 & -0.41869 & -0.08131 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow$$
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.402 \\ 1 & -0.34413 & -0.71414 \\ 1 & 1.772 & 0 \end{bmatrix} \begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} \tag{1}$$

Following [4], to distinguish between actual perception and it's computer representation, we use the term luma for the color space component representing image intensity perception (actual luminance), and term chroma for remaining components responsible for image chrominance.

It is an interesting fact, that analog color space transformation resulting in single luminance and 2 chrominance components is performed by the human vision system. Three types of cone cells in our retinas are most sensitive to three light wavelengths, these are L-cones (long wavelength with sensitivity peak in yellow), M-cones (middle, peak in green) and S-cones (short, peak in violet). Note, that the popular opinion, according to which cones simply respond to red (L-), green (M-) and blue (S-cones) light, is wrong — not only because cone sensitivity peaks are outside of red and blue wavelengths, but also since M- and L-cones are sensitive to the full visible spectrum; S-cones to colors ranging from violet to green. However, the highest reaction to blue color, among all cone types, is indeed shown by S-cones, to green by M-cones, and to red by L-cones. The cone response is then transformed and three calculated components are transmitted to the brain via the optic nerve:

- the luminance being a sum of L- and M-cones response,
- the red minus green color component (a difference between responses of L- and M-cones),
- and the blue minus yellow color component (a difference between response of S-cones and a sum of L- and M-cones responses; it may also be seen as difference between response of S-cones and the luminance).

We mentioned only certain aspects of human color vision reduced to essentials, for thorough description the Reader is referred to [6].

In case of lossless color image compression, the color space transformation has to be reversible considering that transformed components are stored using integers (it has to be integer-reversible). The transformation to the YCbCr color space could be used for that purpose at the cost of a dynamic range expansion of all the transformed color space components by 2 bits [7]. Here, the dynamic range of a component is defined as a number of bits required to store pixel intensities of this component. Since transformations designed for the lossless compression result in better lossless ratios as well as in smaller dynamic range expansion and are of smaller computational complexities, they are used instead. There are several established and standard such transformations, usually being variants of an irreversible transformation. In JPEG2000 for lossless coding the reversible RCT transformation is used [5], it is defined as a series of integer-reversible steps:

$$\begin{array}{lcl} Cv = R - G & & G = Y - \lfloor (Cu + Cv)/4 \rfloor \\ Cu = B - G & \Longleftrightarrow & R = Cv + G \\ Y = G + \lfloor (Cu + Cv)/4 \rfloor & & B = Cu + G \end{array} \tag{2}$$

where the floor symbol $\lfloor x \rfloor$ denotes the greatest integer not exceeding x.

The RCT transformation is computationally simple. Floor of division by integer power of 2 may be calculated using single bit-shift, so both forward and inverse transformations require 5 simple integer operations (add, subtract, bit-shift) per image pixel. The dynamic range of the luma component Y is the same as of RGB components, chroma Cu and Cv components are 1 bit greater.

The RCT transformation was obtained using a lifting scheme [8] for factorization of the below transformation matrix (Eq. (3)) into lifting steps. Hence the below matrix, without additional assumptions, is a close approximation of the RCT transformation only.

$$\begin{bmatrix} Y \\ Cu \\ Cv \end{bmatrix} \approx \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & -1/4 & 3/4 \\ 1 & -1/4 & -1/4 \\ 1 & 3/4 & -1/4 \end{bmatrix} \begin{bmatrix} Y \\ Cu \\ Cv \end{bmatrix} \tag{3}$$

The necessary and sufficient condition for such factorization is that the determinant of the transformation matrix is 1 or -1 [9]. Therefore linear transformations may be made reversible using the lifting scheme with additional scaling of transformation matrix rows if necessary. Notice, that the forward RCT transformation matrix is an approximation of the ICT matrix with scaled chroma rows.

Another such transformation (YCoCg-R) is included in the JPEG XR recent standard [10]. It is a variant of the irreversible transformation (YCoCg), which was obtained based on the KLT transformation constructed for a Kodak set of images (the set is described in Section 3.2) [4]. YCoCg-R is performed in following steps:

$$\begin{array}{lcl} Co = R - B & & t = Y - \lfloor Cg/2 \rfloor \\ t = B + \lfloor Co/2 \rfloor & & G = Cg + t \\ Cg = G - t & \Longleftrightarrow & B = t - \lfloor Co/2 \rfloor \\ Y = t + \lfloor Cg/2 \rfloor & & R = B + Co \end{array} \tag{4}$$

Both forward and inverse transformations require 6 simple integer operations per image pixel. The dynamic range of the luma component Y is the same as of RGB components, chroma Co and Cg components are 1 bit greater. The YCoCg-R transformation approximate forward and inverse matrix equivalents are presented in below Eq. (5).

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} \approx \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & 1/2 & -1/2 \\ 1 & 0 & 1/2 \\ 1 & -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} \tag{5}$$

Sometimes the dynamic range expansion is not allowed or undesirable. Expansion may be not allowed if implementation we use limits the bit-depth of processed data — some implementations allow bit depths not exceeding 8 or 16 bits. Such expansion is undesirable if it involves extra cost, e.g., certain implementations are optimized for 8-bit components, which are processed faster and requiring less memory, than while compressing components of 9–16 bit depth. Expansion may be avoided by means of the modular-arithmetic, the transformation included in the JPEG-LS extended standard (mRCT) is a modular-arithmetic version of the RCT [11]:

$$\begin{array}{lcl} mCv = (R - G) \text{ smod } 2^N & & G = (mY - \lfloor (mCu + mCv)/4 \rfloor) \bmod 2^N \\ mCu = (B - G) \text{ smod } 2^N & \Longleftrightarrow & R = (mCv + G) \bmod 2^N \\ mY = (G + \lfloor (mCu + mCv)/4 \rfloor) \bmod 2^N & & B = (mCu + G) \bmod 2^N \end{array} \tag{6}$$

where N is the dynamic range of components of RGB image expressed in bits per pixel, e.g., for N bit dynamic range the allowable R, G, and B pixel values are in the range $[0\ldots2^N-1]$; $a \bmod 2^N$ is the positive reminder of the division of a by 2^N, or practically the N least significant bits of a, which is in range $[0\ldots2^N-1]$; a smod b is the symmetrical modulo in the range $[-b/2\ldots b/2{-}1]$, a smod $b = ((a+b/2) \bmod b) - b/2$. We assume that both mod and smod are of a similar complexity as, e.g., integer addition. Actually the mod may be simpler (especially for 8 and 16 bit data, where it is just a 1 or 2 Byte memory read). The smod requires two additions, or less, e.g., if pixels are stored as unsigned integers (we may skip subtracting the half of the range). Notice, that not all operations are performed using modular arithmetic, $(mCu + mCv)/4$ is computed in regular arithmetic. The smod is used for chroma components since chroma pixel value, being a difference between primary RGB colors, is most often close to 0. Due to modulo clipping the regular mod would result in creation of greater number of sharp edges in the transformed chroma components. Such edges result from mod on neighboring chroma component pixels close to 0 (between ones smaller than 0 and ones greater or equal 0), while after smod two less frequent cases introduce edges: pixels close to 2^{N-1} and close to $-(2^{N-1})$. Since luma (mY) is calculated using transformed chroma components, also this component contains extra edges (see Fig. 1). The dynamic range of all the transformed components is the same as of RGB components. Both forward and inverse mRCT transformations require 8 simple integer operations per image pixel.

Using the lifting technique the integer-reversible variants of other linear transformations may be obtained, including the KLT. Method presented in [9,12] decomposes the KLT transformation matrix into a series of Triangular or Single-row Elementary Reversible Matrixes, which are then used for integer-reversible KLT transformation (RKLT). In a general case, for 3 component image pixel both forward and inverse RKLT transformations require 8 additions, 8 multiplications and 4 to 5 roundings. Note, that the above complexity does not include the process of computing the KLT transformation matrix and of decomposing it. The KLT dynamic range expansion is image-dependant, for typical RGB images the first component gets expanded by up to 2 bits, while remaining ones are not expanded.

In this paper we present a couple of new color space transformations for lossless image compression. Our primary aim was to find transformations simpler than the established ones, while not being practically inferior in terms of obtainable compression ratios or the dynamic range expansion. New transformations are constructed based on observations of compression effects of individual transformed color components rather than on approximations of KLT transformation; some of them are inspired by the analog calculations taking place in the human vision system.

The reminder of this paper is organized as follows. In Section 2, supported by preliminary experiments, we propose new transformations. In Section 3 we compare experimentally new transformations with others, including the reversible variant of the optimal KLT. Experiments are performed for 3 sets of test images and for 3 significantly different standard compression algorithms: JPEG-LS, JPEG2000 and JPEG XR. We analyze complexities of transformations, compression ratios obtained for the transformed images and correlation of transformed components. Section 5 summarizes the research.

2. Proposed transformations

2.1. Giving the luminance up

Luma calculated from all components is used in lossy compression as it is a good representation of perceptual luminance, to which human vision is more sensitive than to chrominance; luma typically is encoded with higher quality than chroma, a common practice in lossy compression is to subsample chroma components. In lossless compression we usually compress an image in order just to then decompress the entire image. In such a case a good direct approximation of the luminance is not necessary in the transformed image color space. Replacing the luma component with something else may simplify the color space transformation — since, e.g., calculation of the Y component in RCT is more complex than calculation of remaining components. Furthermore, luma calculated based on all components contains a fraction of the noise from all components. As opposed to lossy algorithms that filter out the noise, lossless algorithms preserve all the information contained in the component, including the incompressible noise. Most reversible transformations for lossless compression are based on irreversible transformations for lossy compression, but maybe a different criteria should be applied when constructing transformation for lossless and for lossy compression? Preliminary experiments (Table 1) showed, that for the Waterloo set of test images and the JPEG2000 algorithm in the lossless mode (sets, algorithms and experimental procedure are described in Section 3.2) the untransformed R component, indeed, is more compressible than Y component of the RCT color space, which in turn compresses better, than untransformed components G and B. Note that only certain combinations of three transformed single component formulas from Table 1 constitute the reversible color space transformation (the transformation matrix determinant should be 1 or -1), and that certain transformations are defined using

Fig. 1. Luma components of the "peppers" image after RCT (left) and mRCT (right) transformations.

Table 1
Average JPEG2000 lossless compression ratios of individual transformed components of images from Waterloo set (algorithms and test images described in Section 3.2).

Description	Formula	Ratio (bpp)	Formula	Ratio (bpp)	Formula	Ratio (bpp)
Untransformed	R	4.2562	G	4.3954	B	4.3456
RCT Y variants	$(2R+G+B)/4$	4.2786	$(R+2G+B)/4$	4.3205	$(R+G+2B)/4$	4.3114
Two primary color difference	$R-G$	3.4148	$G-B$	3.4718	$B-R$	3.6532
Negated difference	$G-R$	3.4043	$B-G$	3.4784	$R-B$	3.6506
Average of two primary colors	$(R+G)/2$	4.3112	$(G+B)/2$	4.3533	$(B+R)/2$	4.2867
$B-L$ variants	$R-(G+B)/2$	3.4515	$G-(R+B)/2$	3.2453	$B-(R+G)/2$	3.4937

lifting steps, so the actual ratio of Y of RCT may slightly differ from the ratio reported for component formula $(R+2G+B)/4$.

RCT chroma components seem to be a good choice, since their calculation is done in only one integer subtraction per component pixel and they compress significantly better than Y or untransformed primary color components. It is better to subtract primary colors of closer wavelengths ($R-G$ and $G-B$), then the most distant ones ($B-R$). Therefore in the color space named RDgDb (Eqs. (7) and (8)) we use the R component of the RGB space instead of luminance and, similarly to RCT, use differences of primary colors as chroma components:

$$\begin{bmatrix} R \\ Dg \\ Db \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} R \\ Dg \\ Db \end{bmatrix} \tag{7}$$

Forward and inverse transformations require 2 integer subtractions only:

$$\begin{array}{lll} R = R & & R = R \\ Dg = R - G & \Longleftrightarrow & G = R - Dg \\ Db = G - B & & B = G - Db \end{array} \tag{8}$$

The dynamic range of Dg and Db chroma components is by 1 bit greater, than the dynamic range of RGB components. As opposed to most established color space transformations, where matrix representation is an approximation of actual transformation performed in a series of lifting steps, in case of the RDgDb the matrix is an equivalent definition of the transformation.

Our Db component is a negation of RCT Cu component. We subtract shorter wavelength primary color from one of longer wavelength, while in RCT always the green is subtracted from the other one. The difference is practically not important (negating chroma components would change average ratios by about 0.01 bpp or less), may be of opposite sign for other images or for other algorithms; the formula used clearly suggests how to extend the RDgDb color space to multispectral images.

Transformations close to RDgDb have been recently investigated in [2] — among others, 3 transformations were proposed (A2, A6 and A7), in which a primary color is used instead of luminance while both chroma components are differences in which the primary color replacing luminance is used as subtrahend. Below we present forward transformations A2 (Eq. (9)), A6 (Eq. (10)) and A7 (Eq. (11)).

$$\begin{bmatrix} Y_{A2} \\ U_{A2} \\ V_{A2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{9}$$

$$\begin{bmatrix} Y_{A6} \\ U_{A6} \\ V_{A6} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 1 \\ -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{10}$$

$$\begin{bmatrix} Y_{A7} \\ U_{A7} \\ V_{A7} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{11}$$

Among them the best results were reported for A2, where G is used instead of luminance. However, results presented in the Table 1 indicate, that it is better to use R instead of luminance, while out of two chroma components, both being primary color differences, only in one the component R replacing luminance should be used. Complexity and dynamic range expansion of A2, A5 and A7 is the same as of RDgDb.

2.2. Human vision inspired transformations

If we expect that the luma component may be decompressed without decompressing image chroma components, then a reasonably good approximation of luminance is needed. May the luminance approximation be better than a single primary color, but simpler to compute then the luma component in RCT? The affirmative answer to the above question is in our (human) vision system, where luminance is a sum of responses of two cone cell types: L- and M-cones. Therefore in LDgEb space the luma is, as luminance in humans, a sum of two longer wavelength components, but multiplied by the smallest factor that permits transformation reversibility $L=(R+G)/2$. Following the human vision also for chroma components leads to chroma formulas $Dg=R-G$ and $Eb=B-L$ (Eb is the excess of blue over the luminance). The approximate matrix definition of the LDgEb transformation is:

$$\begin{bmatrix} L \\ Dg \\ Eb \end{bmatrix} \approx \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1 & -1 & 0 \\ -1/2 & -1/2 & 1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & 1/2 & 0 \\ 1 & -1/2 & 0 \\ 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ Dg \\ Eb \end{bmatrix} \tag{12}$$

The integer reversible transformation is defined as a following sequence of steps:

$$\begin{array}{lll} Dg = R - G & & R = L + \lfloor Dg/2 \rfloor \\ L = R - \lfloor Dg/2 \rfloor & \Longleftrightarrow & G = R - Dg \\ Eb = B - L & & B = Eb + L \end{array} \tag{13}$$

Alternatively we may use "human vision" luma, but define chroma components as in RDgDb — obtaining transformation named LDgDb (Eq. (15)). Based on the Table 1 we may expect good compression ratios from both LDgEb and LDgDb transformations. LDgDb actually differs from LDgEb in calculation of one component only:

$$\begin{bmatrix} L \\ Dg \\ Db \end{bmatrix} \approx \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \Longleftrightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & 1/2 & 0 \\ 1 & -1/2 & 0 \\ 1 & -1/2 & -1 \end{bmatrix} \begin{bmatrix} L \\ Dg \\ Db \end{bmatrix} \tag{14}$$

$$\begin{array}{ll} Dg = R - G & R = L + \lfloor Dg/2 \rfloor \\ L = R - \lfloor Dg/2 \rfloor \iff & G = R - Dg \\ Db = G - B & B = G - Db \end{array} \tag{15}$$

Both LDgEb and LDgDb forward and inverse transformations require 4 simple integer operations only. The dynamic range of chroma components is by 1 bit greater, than the dynamic range of RGB components, luma component is not expanded.

2.3. Modular arithmetic transformations

Since avoiding the expansion may be required in practical applications, we define modular-arithmetic versions of RDgDb (mRDgDb), LDgEb (mLDgEb), and of LDgDb (mLDgDb); mRDgDb is performed in following reversible steps:

$$\begin{array}{ll} R = R & R = R \\ mDg = (R - G) \text{ smod } 2^N \iff & G = (R - mDg) \bmod 2^N \\ mDb = (G - B) \text{ smod } 2^N & B = (G - mDb) \bmod 2^N \end{array} \tag{16}$$

where N is a dynamic range of RGB and of transformed mRDgDb components. Forward and inverse mRDgDb transformations require 4 simple integer operations only, while mLDgEb and mLDgDb (also forward and inverse) avoid expansion at a cost of 7 such operations; mLDgEb is performed in following steps:

$$\begin{array}{ll} mDg = (R - G) \text{ smod } 2^N & R = (mL + \lfloor mDg/2 \rfloor) \bmod 2^N \\ mL = (R - \lfloor mDg/2 \rfloor) \bmod 2^N \iff & G = (R - mDg) \bmod 2^N \\ mEb = (B - mL) \text{ smod } 2^N & B = (mEb + mL) \bmod 2^N \end{array} \tag{17}$$

3. Experimental results and discussion

3.1. Examined transformations

Properties of the examined transformations are presented in the Table 2. In the comparison we included RCT, YCoCg-R, and mRCT transformations being both simple and standard. We do not include other transformations, that may approximate YCbCr or KLT better, than the above mentioned ones. Instead, as a benchmark for simple transformations and to check how the optimal decorrelation of color space affects compression ratios we report results obtained using the irreversible KLT (not listed in Table 2) and the reversible KLT (RKLT) – in both cases the transformation was constructed for each image individually. Software developed by Janczur within his MSc thesis [13] was used to construct and to apply the RKLT to the tested images. Other transformations where applied using a prepared free software, which may be downloaded from http://sun.aei.polsl.pl/~rstaros/imgtransf/index.html. We report results for the A2, RDgDb, LDgEb, LDgDb as well as their modular-arithmetic variants. mA2 – the modular arithmetic variant of A2 was constructed analogically to mRDgDb, mLDgDb variant of LDgDb analogically to mLDgEb. We also check compression performance of variants of A2, RDgDb, LDgEb, and LDgDb obtained by the use of different primary colors for luma and if necessary for chroma calculation.

3.2. Procedure

Based on preliminary estimation, performed for the popular Waterloo set of color test images and the JPEG2000 algorithm in the lossless mode, we proposed RDgDb and LDgEb reversible color space transformations as well as a couple of their variants. The evaluation of transformations proposed was performed for the following sets of 8-bit RGB test images:

Table 2
Properties of the reversible color space transformations. Complexities reported in simple integer operations per image pixel, except for RKLT (floating-point operations, not including the cost of computing KLT matrix and decomposing it into lifting steps); component dynamic range expansion is the maximum possible, except for RKLT (typical for natural images).

Transformation	Complexity	Component range expansion		Remarks
		Luma/ first	Chroma/ others	
RKLT	20	2	0	See Refs. [7,10]
RCT	5	0	1	Eq. (2), from JPEG2000 standard [3]
YCoCg-R	6	0	1	Eq. (4), from JPEG XR standard [8]
A2	2	0	1	Eq. (9), see Ref. [11]
RDgDb	2	0	1	Eq. (7) and (8)
LDgEb	4	0	1	Eq. (13)
LDgDb	4	0	1	Eq. (15)
mRCT	8	0	0	Eq. (6), from JPEG-LS Extended standard [9]
mA2	4	0	0	Modular arithmetic variant of A2
mRDgDb	4	0	0	Eq. (16)
mLDgEb	7	0	0	Eq. (17)
mLDgDb	7	0	0	Modular arithmetic variant of LDgDb

- Waterloo – the already mentioned set of color images from the University of Waterloo, Fractal Coding and Analysis Group repository, used for a long time in image processing research. The set contains 8 natural photographic and artificial images, among them the well-known "lena" and "peppers", image sizes vary from 512×512 to 1118×1105. It is available from http://links.uwaterloo.ca/Repository.html.
- Kodak – a set of 24 photographic images released by the Kodak corporation, the set is frequently used in color image compression research. All images are of size 768×512, downloaded from http://www.cipr.rpi.edu/resource/stills/kodak.html.
- EPFL – a recent set of 10 high resolution images used at the École polytechnique fédérale de Lausanne for evaluation of subjective quality of JPEG XR [14]. Image sizes from 1280×1506 to 1280×1600, downloaded from http://documents.epfl.ch/groups/g/gr/gr-eb-unit/www/IQA/Original.zip.

Lossless compression ratios obtained for the transformed images were analyzed for the following standard algorithms:

- JPEG-LS – a standard of the JPEG committee for primarily lossless compression of still images. The baseline standard describes low-complexity predictive image compression algorithm with entropy coding using modified Golomb-Rice code family, standard extensions include the mRCT transformation [11,15].
- JPEG2000 – a JPEG committee image compression standard describing algorithm based on discrete wavelet transformation image decomposition and arithmetic coding, the standard includes the RCT color space transformation [5]. Apart from lossy and lossless compressing and decompressing of whole images JPEG2000 delivers many interesting features (progressive transmission, region of interest coding, etc.).
- JPEG XR – a recent JPEG committee standard describing algorithm designed primarily for high quality, high dynamic range photographic images; it is based on discrete cosine transformation image decomposition and adaptive Huffman coding; it defines the YCoCg-R color space transformation [10]. The standard supports lossy and lossless coding and certain additional features, like random access to parts of the encoded image.

Above algorithms process images in significantly different ways. First step of color image compression is similar: using the color space transformation we obtain 3 components, that are then compressed independently. In a predictive algorithm the next step for each component is to use the predictor function to guess the pixel intensities and then the sequence of prediction errors, being differences between actual and predicted pixel intensities, is encoded. Even using extremely simple predictors, such as one that predicts that pixel intensity is identical to the one on its left-hand side, results in a much better compression ratio, than without the prediction. JPEG-LS employs nonlinear edge-detecting predictor calculated using 4 neighbors of given pixel. In transformation algorithms, instead of pixel intensities, we encode a matrix of transformation coefficients. The transformation is applied to a whole image component (optional in JPEG2000), or to the component split into fragments, which may be big (default fragment size for JPEG2000 is 256×256 pixels), or even very small — JPEG XR applies transformation to blocks of size 4×4 pixels (2×2 in case of lossy chroma component coding). Transformation for lossless coding has to be integer-reversible, which for JPEG2000 and JPEG XR is accomplished with use of the lifting scheme.

All algorithms were used to compress individual transformed components, one component at a time. Due to requirements of employed file formats and standard implementations, all transformed components were stored using nonnegative integers obtained by subtracting the minimum nominally possible value from the pixel component. For example, if primary colors were in nominal range $[0, 255]$, then $Dg = R - G$, which would normally be in the range $[-255, 255]$, was actually stored as $Dg' = R - G + 255$. The compression ratio, expressed in bits per pixel (bpp), is calculated as $8e/n$, where n is the number of pixels in the image, e is the total size in Bytes of the individually and independently compressed 3 components of the transformed image, including compressed file format headers; hence smaller ratios mean better compression. We also report average absolute correlation of image components after transformation, which is calculated as $(|r(C_1, C_2)| + |r(C_2, C_3)| + |r(C_3, C_1)|)/3$, where C_1, C_2, and C_3 are components of the transformed image, r is Pearson product moment correlation coefficient, and $|x|$ is the absolute value of x.

3.3. Lossless compression ratios

Lossless compression ratios for transformed images are reported in Tables 3 (for JPEG-LS), 4 (JPEG2000) and 5 (JPEG XR). The best transformation result, for a given set and for average of 3 sets, is marked in bold; underlined are results not worse by more than 0.1 bpp then the best. Except for the transformations listed in the Table 2, we include results for irreversible KLT (irrevKLT).

Looking at the results of RKLT and irrevKLT transformations we can see, that optimal decorrelation of the image color space, even in its irreversible variant, does not lead to the best lossless ratios. Compared to the oldest reversible transformation RCT, for all the algorithms and all the sets, the RKLT and irrevKLT transformations resulted in ratios worse by at least about 0.5 bpp and 0.3 bpp respectively.

Among modular-arithmetic transformations, the best ratios for all the sets and for all the algorithms were obtained using the mRDgDb, except for the Kodak set in case of JPEG-LS (where the mLDgEb is by 0.06 bpp better). Modular-arithmetic transformations, as compared to their regular variants, may improve the compression ratios only in the case of JPEG-LS — ratios are improved for mRDgDb and mA2 in case of all the sets, and for all the transformations in case of the Kodak set; for JPEG2000 and JPEG XR ratios are consistently worsened. The improvement over regular variant always is the greatest in case of mRDgDb and mA2, also for these transformations the ratio worsening due to modulo arithmetic for JPEG2000 and JPEG XR is the smallest. Probable reason of the above advantage of the simplest among modular transformations over others is that all image components after other transformations contain edges introduced by modulo clipping, while after mRDgDb and mA2 only chroma components contain extra edges. Since JPEG-LS obtains better lossless ratios than other two algorithms, the overall best average ratios were obtained using mRDgDb and JPEG-LS.

The best ratios average for 3 sets, among all the transformations in case of JPEG2000 and JPEG XR as well as among non-modular transformations in case of JPEG-LS, are obtained using the simplest RDgDb. Non-modular transformations using human vision inspired luma component formula obtain ratios worse by up to 0.04 bpp, A2 is worse by 0.08 to 0.09 bpp, standard transformations RCT and YCoCg-R are worse by 0.05 to 0.15 bpp. RDgDb is most often the best transformation for a specific set. However, in case of the JPEG XR algorithm the YCoCg-R is better for Waterloo and Kodak sets (for Waterloo it obtains the best ratio). For the Kodak set in case of all the algorithms either LDgEb or LDgDb is the best non-modular transformation, the other one is the second best — notice that worse ratios were obtained by YCoCg-R based on KLT constructed for this set, the worst by KLT variants constructed for each image individually.

Since the proposed color spaces were constructed based on estimation of ratios for Waterloo set and JPEG2000 algorithm only, for all the sets and algorithms we checked their additional variants as well as variants of A2 transformation. For transformations which replace luminance with a primary color and require only 2 simple integer operations per pixel: RDgDb and A2, there are 9 possible variants not counting cases when given component is calculated as difference or negated difference. There are 3 variants of A2 (having following components: a primary color as luma and two differences between another primary colors and luma) and 6 variants of RDgDb (components: a primary color as luma, difference between luma and another primary color, difference between two primary colors other than luma). Out of these 9 variants the best for a specific set and on average was RDgDb, except for the Waterloo set, where for all the algorithms ratios better by less than 0.02 bpp were obtained when green color replaced luminance. Among variants of transformations requiring 4 operations: LDgEb (3 variants) and LDgDb (6 variants) either LDgEb or LDgDb was the best, except for the Waterloo set in case of JPEG2000 and JPEG XR, where LDgEb variant using R and B primary colors in luma calculation was better than LDgEb by about 0.04 bpp.

In practice, small differences in average compression ratios may not be the most important property of the color space transformation. Properties like dynamic range expansion, computational complexity, quality of luminance approximation, and existence of the standard describing the transformation may play practical role. In Tables 3–5 the ratios within 0.1 bpp from the best one are underlined. The selection of the 0.1 threshold was arbitrary, but observing when the ratios are within 0.1 bpp from the best generally confirms earlier observations: consistently outside of that range on average and for all the sets are modular-arithmetic transformations in case of JPEG2000 and JPEG XR, as well as KLT variants in all cases. For JPEG-LS the non-modular RCT and YCoCg-R are always outside. For all the sets the 3 proposed non-modular transformations are in 0.1 bpp range in case of JPEG2000 and JPEG XR compression. Other cases when for all the sets given transformation is within 0.1 bpp from the best one are RCT for JPEG XR, and mRDgDb for JPEG-LS. Looking at the average ratios, the proposed non-modular transformations are in the range for all the algorithms, while other transformations are for some only or for none.

Table 3
Average JPEG-LS compression ratios (bpp). Best ratio for a set and on average is marked in bold, ratios within 0.1 bpp from the best are underlined.

Transformation	Set			
	Waterloo	Kodak	EPFL	Average
No (RGB)	10.3764	13.0948	12.3369	11.9360
RKLT	9.5987	10.4365	11.0722	10.3691
irrevKLT	9.4989	10.1976	10.8014	10.1659
RCT	8.9625	9.5734	10.4660	9.6673
YCoCg-R	9.0232	9.6044	10.6125	9.7467
A2	8.9914	9.5502	10.4930	9.6782
RDgDb	8.8653	9.5673	10.3383	9.5903
LDgEb	8.9589	9.4335	10.4421	9.6115
LDgDb	8.9309	9.5476	10.4181	9.6322
mRCT	9.0017	9.4805	10.5079	9.6633
mA2	8.9546	9.4387	10.4719	9.6217
mRDgDb	**8.8285**	9.4559	**10.3172**	**9.5338**
mLDgEb	9.1277	**9.3985**	10.4992	9.6751
mLDgDb	8.9880	9.4580	10.4338	9.6266

Table 4
Average JPEG2000 compression ratios (bpp). Best ratio for a set and on average is marked in bold, ratios within 0.1 bpp from the best are underlined.

Transformation	Set			
	Waterloo	Kodak	EPFL	Average
No (RGB)	12.9972	13.4256	12.8244	13.0824
RKLT	11.8353	10.5579	11.3968	11.2634
irrevKLT	11.7252	10.3329	11.1456	11.0679
RCT	11.2141	9.5062	10.8356	10.5186
YCoCg-R	11.2125	9.4876	10.9776	10.5593
A2	11.2819	9.4686	10.8655	10.5387
RDgDb	**11.1428**	9.4754	**10.7240**	**10.4474**
LDgEb	11.2178	**9.4231**	10.8054	10.4821
LDgDb	11.1969	9.4590	10.7839	10.4799
mRCT	11.6473	9.6107	11.0823	10.7801
mA2	11.5023	9.5375	10.9883	10.6760
mRDgDb	11.3631	9.5443	10.8468	10.5848
mLDgEb	11.8572	9.5766	11.0409	10.8249
mLDgDb	11.6436	9.5652	10.9711	10.7266

Table 5
Average JPEG XR compression ratios (bpp). Best ratio for a set and on average is marked in bold, ratios within 0.1 bpp from the best are underlined.

Transformation	Set			
	Waterloo	Kodak	EPFL	Average
No (RGB)	14.9604	14.1331	13.6662	14.2532
RKLT	13.8199	11.6542	12.3010	12.5917
irrevKLT	13.7459	11.5089	12.1505	12.4684
RCT	13.3211	10.9196	11.7629	12.0012
YCoCg-R	**13.2655**	10.8552	11.8987	12.0065
A2	13.4243	10.8565	11.7911	12.0240
RDgDb	13.2978	10.8725	**11.6723**	**11.9476**
LDgEb	13.2982	10.8531	11.7402	11.9638
LDgDb	13.3108	**10.8484**	11.7154	11.9582
mRCT	14.2161	11.1797	12.2143	12.5367
mA2	13.9917	11.0834	12.0583	12.3778
mRDgDb	13.8652	11.0995	11.9395	12.3014
mLDgEb	14.5268	11.1971	12.1715	12.6318
mLDgDb	14.2450	11.1296	12.0795	12.4847

3.4. Component correlation

The average absolute correlation of image components, in RGB color space and after the examined transformations, is reported in Table 6. We can see, that both the irreversible KLT and its integer-reversible lifting approximation decorrelate image color space almost perfectly, which confirms observation that decorrelation of the color space is not a proper objective of the color space transformation for lossless compression. The opposite may be true — greater correlation of image components may potentially allow for better compression in algorithms that during compression of given component exploit other, already processed ones. There are algorithms, which instead of component transformation, exploit during coding the inter-component dependencies, e.g., the relatively time complex Interband CALIC [16]; low complexity algorithms were also proposed, e.g., SICLIC [17] or recent LMMIC [18]. Interband CALIC uses the inter-component predictor instead of regular (intra-component) one if in the neighborhood of the pixel being predicted the correlation between components is high. For such an algorithm, out of two transformations which would result in equal ratios for independently compressed components, probably better is the transformation decorrelating worse. However checking whether is it better than not to transform and leave all the correalation in components requires further investigation.

Table 6
Average absolute correlation of transformed image components.

Transformation	Set			
	Waterloo	Kodak	EPFL	Average
No (RGB)	0.6311	0.8435	0.8002	0.7583
RKLT	0.0005	0.0019	0.0009	0.0011
irrevKLT	0.0024	0.0053	0.0046	0.0041
RCT	0.3187	0.3100	0.3058	0.3115
YCoCg-R	0.2616	0.3374	0.3507	0.3165
A2	0.3978	0.3451	0.3095	0.3508
RDgDb	0.4006	0.3496	0.3455	0.3652
LDgEb	0.3746	0.3899	0.3972	0.3872
LDgDb	0.3633	0.3269	0.2980	0.3294
mRCT	0.1974	0.2870	0.2944	0.2596
mA2	0.1932	0.2972	0.2741	0.2548
mRDgDb	0.2288	0.3074	0.3120	0.2827
mLDgEb	0.2894	0.3587	0.3443	0.3308
mLDgDb	0.2075	0.2953	0.2715	0.2581

Interestingly, while the compression ratio of irrevKLT is better than of RKLT, the correlation of components is little greater in case of irrevKLT. Probably irrevKLT due to rounding removes fraction of noise contained in the image. Presence of noise on the one hand worsens compression ratios, on the other one decreases correlation of components.

Among non-modular transformations, ones being approximations of KLT (i.e., RCT and YCoCg-R) on average decorrelate better, than transformations constructed based on different criteria (A2, RDgDb, LDgEb, and LDgDb), however for a specific set it may not be true. In all cases modular-arithmetic transformations decrease correlation more, than corresponding regular variants. On average LDgEb decorrelates worstly, for a specific set — LDgEb or RDgDb.

4. Conclusions

We have proposed RDgDb and LDgEb simple color space transformations for lossless image compression and a couple of their variants. We departed from a traditional method of constructing transformation for lossless image compression based on transformation for lossy compression, which in turn is based on PCA/KLT for specific image set. RDgDb was proposed based on observation of actual lossless ratios of individual image components obtained with simple transformations or untransformed, while LDgEb originates from the human vision system. These transformations were evaluated and compared with established transformations including RCT, YCoCg-R and the optimal KLT for 3 sets of test images and for significantly different compression algorithms: predictive JPEG-LS, Discrete Wavelet Transformation based JPEG2000 and Discrete Cosine Transformation based JPEG XR.

The RDgDb transformation has the minimum computational complexity, equal to complexity of the simplest known so far A2 transformation, it requires just 2 integer subtractions per image pixel. RDgDb and A2 have certain disadvantages: the dynamic range of chroma components is expanded by 1 bit, the luminance is replaced by a primary color. However in a typical case of lossless compression, when an image is compressed just to be then decompressed as a whole, RDgDb seems to be the most universal — despite of being so simple, on average it results in the best ratios for JPEG2000 and JPEG XR, while for a specific set or in case of JPEG-LS its compression ratios are either the best or within 0.1 bpp from the best.

The overall best lossless ratios were obtained using the JPEG-LS algorithm and the modular-arithmetic variant of RDgDb, named mRDgDb, which requires 2 integer subtractions and 2 symmetrical modulo operations. Here RDgDb resulted in second best average ratio, worse by about 0.06 bpp. In practice such a little ratio improvement may not justify the increased complexity of the transformation. As all modular arithmetic transformations the mRDgDb avoids the dynamic range expansion of image components; compared to them it is of the lowest complexity and results in the best average ratios for all the algorithms. It will be practically useful in cases where the dynamic range expansion is not allowed or undesirable.

Sometimes a reasonably good perceptual luminance approximation in a transformed image may be useful, since it allows retrieving the luminance from compressed image by decompressing of one component only. We proposed LDgEb and LDgDb transformations inspired by analog calculations performed in human vision system, which in the above case may be useful for natural images. Compared to RDgDb these transformations are more complex and result in little worse average ratios, but compared to RCT and YCoCg they are simpler and result in better ratios. The most interesting is LDgEb, since it is closest to transformations in human vision system and for one of the test image sets it obtains the best ratios (its modular arithmetic variant in case of JPEG-LS).

We also notice that optimal color space decorrelation performed with KLT, despite of constructing the KLT for each image individually and even giving up the integer reversibility, does not lead to good lossless ratios. Color space decorrelation is not a proper aim of transformation for lossless compression, controversially poor decorrelation may allow for better compression in algorithms exploiting inter-plane correlation. In the case of tested transformations, the greatest correlation of transformed image components is observed for LDgEb and RDgDb.

Color space components after the LDgEb transformation are closer to components transmitted to the human brain via the optic nerve, than components of spaces traditionally used in color image digital transmission, like YCbCr or untransformed RGB. On the other hand, YCbCr and RGB spaces are used directly in various algorithms that generally are aimed at mimicking effects of image processing and analysis conducted by the human vision system (e.g., image retrieval and recognition). Checking whether by employing LDgEb the results of such algorithms will get closer to results we expect from experience with our own visual system is an interesting field of future research. Naturally, for above applications we do not need integer reversibility of the transformation, so LDgEb chroma components may be scaled down by a factor of 2 making the transformed color space components dynamic range equal to the range of untransformed components. Other potential fields of further research are: extending RDgDb to multispectral data and applying LDgEb, or it's above-mentionedvariant with scaled chroma rows, to lossy compression.

Conflict of interest statement

None.

Role of the funding source

The study sponsors were not involved in the study design, in the collection, analysis and interpretation of data, in the writing of the manuscript, or in the decision to submit the manuscript for publication.

Acknowledgments

This work was supported by BK-219/RAu2/2012 grant from the Institute of Computer Science, Silesian University of Technology and by POIG.02.03.01-24-099/13 grant: GCONiI – Upper-Silesian Center for Scientific Computations.

References

[1] W.K. Pratt, Digital Image Processing, Wiley, New York, 1978.
[2] T. Strutz, Adaptive selection of colour transformations for reversible image compression, in: Proc. of the 20th European Signal Processing Conference (EUSIPCO), 2012, pp. 1204–1208.
[3] S.K. Singh, S. Kumar, Novel adaptive color space transform and application to image compression, Signal Process.: Image Commun. 26 (10) (2011) 662–672, http://dx.doi.org/10.1016/j.image.2011.08.001.
[4] H.S. Malvar, G.J. Sullivan, S. Srinivasan, Lifting-based reversible color transformations for image compression, Proc. SPIE 7073 (2008) 707307, http://dx.doi.org/10.1117/12.797091.
[5] ISO/IEC and ITU-T, Information Technology – JPEG2000 image coding system: core coding system, ISO/IEC International Standard 15444-1 and ITU-T recommendation T.800, 2004.
[6] K.R. Gegenfurtner, D.C. Kiper, Color vision, Annu. Rev. Neurosci. 26 (2003) 181–206, http://dx.doi.org/10.1146/annurev.neuro.26.041002.131116.
[7] M. Domański, K. Rakowski, Lossless and near-lossless image compression with color transformations, Proc. ICIP 3 (2001) (2001) 454–457, http://dx.doi.org/10.1109/ICIP.2001.958149.
[8] I. Daubechies, W. Sweldens, Factoring wavelet transforms into lifting steps, J. Fourier Anal. Appl. 4 (1998) 247–269.
[9] P. Hao, Q. Shi, Matrix factorizations for reversible integer mapping, IEEE Trans. Signal Process. 49 (10) (2001) 2314–2324, http://dx.doi.org/10.1109/78.950787.
[10] ISO/IEC and ITU-T, JPEG XR Image Coding Specification, ISO/IEC International Standard 29199-2 and ITU-T Recommendation T.832, 2009.
[11] ISO/IEC and ITU-T, Information technology – Lossless and near-lossless compression of continuous-tone still images: Extensions, ISO/IEC International Standard 14495-2 and ITU-T Recommendation T.870, 2003.
[12] P. Hao, Q. Shi, Reversible integer KLT for progressive-to-lossless compression of multiple component images, in: Proc IEEE Int. Conf. Image Process, 2003, pp. 633–636.
[13] P. Janczur, Reversible PCA/KLT transformation in color image compression, M.Sc. thesis under supervision of R. Starosolski, Institute of Computer Science, Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Gliwice, 2011.
[14] F. De Simone, L. Goldmann, V. Baroncini, T. Ebrahimi, Subjective evaluation of JPEG XR image compression, Proc. SPIE 7443 (2009) 74430L, http://dx.doi.org/10.1117/12.830714.
[15] ISO/IEC and ITU-T, Information technology – Lossless and near-lossless compression of continuous-tone still images – Baseline, ISO/IEC International Standard 14495-1 and ITU-T Recommendation T.87, 2000.
[16] X. Wu, N. Memon, Context-based lossless interband compression – extending CALIC, IEEE Trans. Image Process. 9 (6) (2000) 994–1001.
[17] R. Barequet, M. Feder, SICLIC: a simple inter-color lossless image coder, in: Proc. Data Compression Conference DCC'99, 1999, pp. 501–510.
[18] X. Chen, N. Canagarajah, J.L. Nunez-Yanez, Lossless multi-mode interband image compression and its hardware architecture. Algorithm-architecture matching for signal and image processing, Lect. Notes Electr. Eng. 73 (2011) 3–26, http://dx.doi.org/10.1007/978-90-481-9965-5_1.