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/

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

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:

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

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.

$$\begin{bmatrix} Y \\ Cu \\ Cv \end{bmatrix} \approx \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \iff \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & -1/4 & 3/4 \\ 1 & -1/4 & -1/4 \\ 1 & 3/4 & -1/4 \end{bmatrix} \begin{bmatrix} Y \\ Cu \\ Cv \end{bmatrix} \quad (3)$$

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:

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

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).

$$\begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} \approx \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1 & 0 & -1 \\ -1/2 & 1 & -1/2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \iff \begin{bmatrix} R \\ G \\ B \end{bmatrix} \approx \begin{bmatrix} 1 & 1/2 & -1/2 \\ 1 & 0 & 1/2 \\ 1 & -1/2 & -1/2 \end{bmatrix} \begin{bmatrix} Y \\ Co \\ Cg \end{bmatrix} \quad (5)$$

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]:

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