CONTENT-ADAPTIVE COLOR TRANSFORM FOR IMAGE COMPRESSION

Alexander Suhre, Kivanc Kose, A. Enis Cetin

Bilkent University
Department of Electrical and
Electronics Engineering
Bilkent University,
06800 Bilkent, Ankara, Turkey

Metin N. Gurcan

The Ohio State University
Department of Biomedical Informatics
3190 Graves Hall, 333 W. Tenth Ave.
Columbus, OH 43210

ABSTRACT

In this paper, an adaptive color transform for image compression is introduced. In each block of the image coefficients of the color transform are determined from the previously compressed neighboring blocks using weighted sums of the original RGB-channels of an image, making the transform block-specific. There is no need to transmit or store the transform coefficients because they are estimated from previous blocks. The compression efficiency of the transform is demonstrated using the JPEG image coding scheme. In general, the suggested transformation results in better PSNR values for a given compression level when tested on 15 commonly used test images.

Index Terms— Image compression, JPEG, color transform

1. INTRODUCTION

Image compression is a well-established and extensively studied field in the signal processing and communication communities. Although the "lossy" JPEG standard [1] is the most widely accepted image compression technique in modern day applications, its optimality and resulting fidelity can be improved. One possible idea is to find a colour transform that represents the RGB components in a more efficient manner and can thereby replace the well-known RGB-to-YCbCr or RGB-to-YUV color transforms, used by most practitioners. A new transform based on the colour content of a given image is developed in the remainder of this paper. The proposed transform can be used as part of the JPEG image coding standard, as well as part of other image and video coding methods, including the methods described in [2] and [3].

2. ALGORITHM

A given colorspace transform can mathematically be represented by a matrix multiplication as follows:

$$\begin{pmatrix} D \\ E \\ F \end{pmatrix} = \mathbf{T} \cdot \begin{pmatrix} R \\ G \\ B \end{pmatrix}, \quad (1)$$

where $\mathbf{T} = [t_{ij}]_{3\times3}$ is the transform matrix, while R, G and B represent the red, green and blue colour components of a given pixel, respectively, and D, E, F represent the transformed values. For example, JPEG uses the RGB-to-YCbCr colorspace transform and chooses the coefficients in $\mathbf{T}$ accordingly. The Y component of the resultant image is usually called the luminance component, carrying most of the information, while the Cb and Cr components are called the chrominance components.

In our approach, we manipulate the luminance component, while leaving the chrominance components as they are, i.e., we only modify the coefficients in the first row of the $\mathbf{T}$-matrix. The second and third rows of the matrix remain unaltered.

We calculate the coefficients t_{11}, t_{12}, t_{13} of the first row of the matrix, using the colour content histogram of the image in the following manner:

$$t_{11} = \frac{1}{2} \cdot \left(t'_{11} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,1)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k,l,m)} \right), \quad (2)$$

$$t_{12} = \frac{1}{2} \cdot \left(t'_{12} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,2)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k,l,m)} \right) \quad (3)$$

and

$$t_{13} = \frac{1}{2} \cdot \left(t'_{13} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i,j,3)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k,l,m)} \right), \quad (4)$$

where $\mathbf{I}$ denotes a three-dimensional, discrete RGB image, M and N denote the number of rows and columns of the image, respectively and t'_{ij} denotes the element in the i-th row and the j-th column of the 3-by-3 baseline colortransform matrix, e.g. RGB-to-YCbCr. Equations (2)-(4) have to be computed for each image, therefore, the proposed transform is specific to the given image. This would mean that an extra overhead for encoding the color transform matrix is needed. This extra overhead can be avoided, however. Consider the following: The original image can be divided into subimages, of variable sizes, e.g. 8-by-8 pixels. These subimages are raster-scanned and are fed to a JPEG encoder one by one. For the first subimage, the baseline colortransform is used and its histogram is computed according to equations (2)-(4). For the second subimage the histogram computed in the previous step is used and its coefficients are inserted into the first row of the baseline color transform. The histogram for the second subimage is computed and used for the third subimage. Due to the raster-scanning, the correlation between neighboring subimages is expected to be high, therefore the histogram of one subimage is supposed to be a sufficiently good estimate of its neighbor's histogram, given the subimage size is chosen reasonably. A more general description of this procedure is shown in Figure 1. The above described algorithm is applied to all subimages. Thus, no additional information on the color transform needs to be encoded.

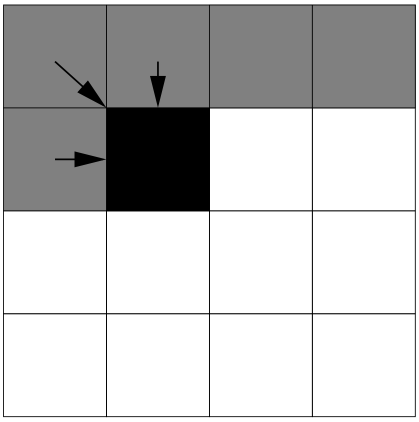

Fig. 1. A general description of our prediciton scheme. To predict the histogram of the black subblock of the image, the histograms of the gray subblocks may be known but only the ones indicated with an arrow are used.

3. EXPERIMENTAL STUDIES AND RESULTS

Three experiments are conducted. For these we use a dataset of 15 images, containing 'Lenna', 'Baboon', 'Goldhill', 'Boats', 'Pepper', 'Airplane', 'Barbara' and some images of natural scenes from the 'Kodak' dataset [4].

In the first experiment, we use JFIF's RGB-to-YCbCr [5], RGB-to-YUV [6] and a digital version of RGB-to-Y'CbCr [7] as our baseline color transforms. Their respective transform matrices are given as

$$\mathbf{T'}_{RGB-to-YCbCr} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix}, \quad (5)$$

$$\mathbf{T'}_{RGB-to-YUV} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix}, \quad (6)$$

and

$$\mathbf{T'}_{RGB-to-Y'CbCr} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix}. \quad (7)$$

We use the JPEG coder, available in MATLAB's *imwrite* function. We replace the color transformation stage of the baseline JPEG with the proposed form of transformation. Results of the experiments are summarised in Tables 1 - 3. For the preparation of these tables, the PSNR-Gain of our method over the baseline color transform are measured at five different rates, spread over the whole rate range, for each image. The averages of these gain values are shown in the tables. Additionally, the mean of all these gain values is presented for the whole dataset. Furthermore, a success rate for the dataset is given. The decision for a success is binary and is made in case the gain value of a given image is positive. These results show that the proposed method produce better results than the baseline JPEG algorithm using the YCbCr, YUV or Y'CbCr matrices, respectively.

Image Name	Average PSNR-Gain [dB]
'1'	0.0917
'2'	-0.2833
'3'	-0.5851
'4'	-0.0377
'5'	0.0343
'6'	0.1177
'7'	0.0226
'8'	0.1356
'Airplane'	0.0114
'Baboon'	0.6837
'Barbara'	-0.0031
'Boats'	0.0786
'Goldhill'	0.2239
'Lenna'	0.3463
'Pepper'	0.2908
Whole dataset	0.0752
Success Rate	11/15

Table 1. PSNR-Gain values for the whole dataset with RGB-to-YCbCr as baseline colour transform. PSNR-Gain of each image is measured at different rates and averaged.

Image	Average PSNR-Gain [dB]
’1’	0.0692
’2’	-0.3555
’3’	-1.1145
’4’	-0.2681
’5’	0.0473
’6’	0.1310
’7’	-0.0190
’8’	0.1141
’Airplane’	0.0013
’Baboon’	0.5735
’Barbara’	-0.6658
’Boats’	0.0337
’Goldhill’	0.1911
’Lenna’	0.3431
’Pepper’	0.2062
Whole dataset	-0.0475
Success Rate	10/15

Table 2. PSNR-Gain values for the whole dataset with RGB-to-YUV as baseline colour transform. PSNR-Gain of each image is measured at different rates and averaged.

Image	Average PSNR-Gain [dB]
’1’	0.2007
’2’	-0.1820
’3’	0.1434
’4’	0.1083
’5’	0.2779
’6’	0.2919
’7’	0.2013
’8’	0.2914
’Airplane’	0.1974
’Baboon’	0.7897
’Barbara’	0.3632
’Boats’	0.2375
’Goldhill’	0.3307
’Lenna’	0.4782
’Pepper’	0.3684
Whole dataset	0.2732
Success Rate	14/15

Table 3. PSNR-Gain values for the whole dataset with RGB-to-Y’CbCr as baseline colour transform. PSNR-Gain of each image is measured at different rates and averaged.

In Figures 2, 3, 4, the *PSNR* vs. *Compression Ratio (CR)* curves for the above mentioned color transforms of the images ’Baboon’, ’Lenna’ and ’Pepper’ are given, respectively.

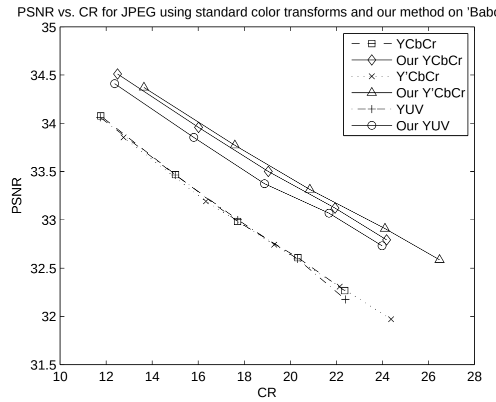

Fig. 2. PSNR-vs-CR performance of the ’Baboon’ image for fixed color transforms and our method.

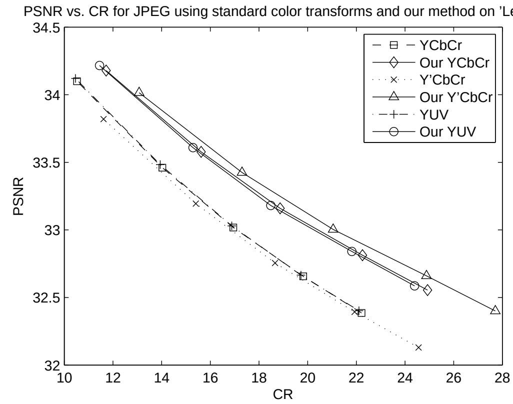

Fig. 3. PSNR-vs-CR performance of the ’Lenna’ image for fixed color transforms and our method.

4. CONCLUSION

We presented a method of extracting an image-specific color transform using a weighted sum-based approach. The first row of the transform matrix basically consists of color component ratios of a given image. Our experiments suggest that when this transform is used in standard JPEG, it results in higher PSNR for a given CR than standard colorspace transforms in general. Due to its conceptual simplicity and computational efficiency, our method can also be used in video compression.

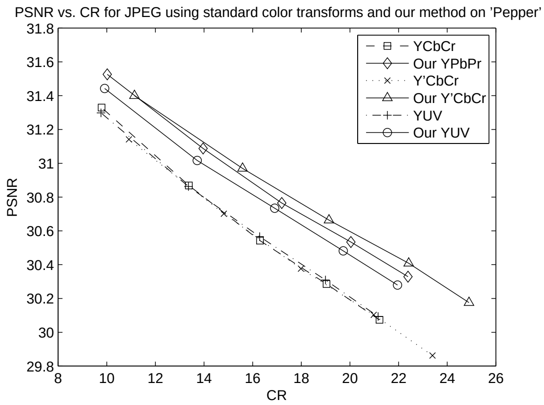

Fig. 4. PSNR-vs-CR performance of the 'Pepper' image for fixed color transforms and our method.

5. REFERENCES

[1] G.K. Wallace, "The JPEG Still Picture Compression Standard", *IEEE Transactions on Consumer Electronics, Vol. 38, No. 1* pp. 18-34, 1992.

[2] O.N. Gerek and A.E. Cetin, "Adaptive polyphase subband decomposition structures for image compression", *Image Processing, IEEE Transactions on , vol.9, no.10* pp.1649-1660, 2000

[3] O.N. Gerek and A.E. Cetin, "A 2-D orientation-adaptive prediction filter in lifting structures for image coding", *Image Processing, IEEE Transactions on , vol.15, no.1* pp.106-111, Jan. 2006

[4] `http://www.xeraina.ch/pgf/pgfdownload.html`

[5] E. Hamilton, "JPEG File Interchange Format (Version 1.02)," C-Cube Microsystems, Milpitas, California, 1992.

[6] `http://en.wikipedia.org/wiki/yuv`

[7] `http://en.wikipedia.org/wiki/ycbcr`

[8] W. K. Pratt, "Spatial transform coding of color images", *IEEE. Trans. Commun., vol. 19* pp. 980-992, 1971.

[9] Y. Ohta, T. Kanade and T. Sakai, "Color Information for Region Segmentation", *Computer Graphics and Image Processing, Vol. 13, No. 3* pp. 222 - 241, 1980.

[10] E. Gershikov and M. Porat, "On color transforms and bit allocation for optimal subband image compression", *Signal Processing: Image Communication, Vol. 22, Issue 1* pp. 1-18, 2007.