Optical Engineering

SPIEDigitalLibrary.org/oe

Content-adaptive color transform for image compression

Alexander Suhre
Kivanc Kose
A. Enis Cetin
Metin N. Gurcan

Content-adaptive color transform for image compression

Alexander Suhre
Kivanc Kose
A. Enis Cetin
Metin N. Gurcan
Bilkent University
Department of Electrical and Electronics Engineering
06800 Ankara, Turkey
and
Ohio State University
Department of Biomedical Informatics
Columbus, Ohio 43210
E-mail: suhre@ee.bilkent.edu.tr

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 RGB pixel values, 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 peak signal-to-noise ratio (PSNR) values for a given compression level. [DOI: 10.1117/1.3574071]

Subject terms: image compression; JPEG; color transform.

Paper 110008R received Jan. 5, 2011; revised manuscript received Mar. 11, 2011; accepted for publication Mar. 14, 2011; published online May 9, 2011.

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 one of the most widely accepted image compression techniques in modern day applications, its resulting fidelity can be improved. It is a well known fact that JPEG compression standard is optimized for natural images. More specifically the color transformation stage is designed in such a way that it favors the color components to which the human visual system is more sensitive in general. However, using one fixed color transformation for all types of natural images or even for all the blocks of an image may not be the most efficient way. One possible idea is to find a color 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. Usually such approaches aim at reducing the correlation between the color channels.[2] An optimal solution would be to use Karhunen–Loève transform (KLT) (see Ref. 3). However, in KLT there is an underlying wide-sense stationary random process assumption which may not be valid in natural images. This is because autocorrelation values of the image have to be estimated to construct the KLT matrix, since most natural images cannot be considered as wide-sense stationary random processes, due to edges and different objects. A single auto-correlation sequence cannot represent a given image. Another approach to an optimal color space projection on a four-dimensional colorspace was developed in Ref. 4.

A new transform based on the color content of a given image is developed in 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 Refs. 5,6, and 7.

2 Algorithm

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

$$\begin{pmatrix} D \\ E \\ F \end{pmatrix} = \mathbf{T} \cdot \begin{pmatrix} \mathrm{R} \\ \mathrm{G} \\ \mathrm{B} \end{pmatrix}, \tag{1}$$

where $\mathbf{T} = [t_{ij}]_{3\times 3}$ is the transform matrix, while R, G and B represent the red, green, and blue color components of a given pixel, respectively, and *D, E, F* represent the transformed values, see Refs. 8 and 9. For example, JPEG uses luminance-chrominance type colorspace transforms and chooses the coefficients in **T** accordingly. Examples for these include RGB-to-YCbCr,[10] as definded in JPEG file interchange format, as well as RGB-to-YUV and a digital version of RGB-to-Y′CbCr from CCIR 601 Standard that are used in our experiments as baseline color transforms. Their respective transform matrices are given by

$$\mathbf{T}'_{\mathrm{RGB-to-YCbCr}} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix}, \tag{2}$$

$$\mathbf{T}'_{\mathrm{RGB-to-Y'CbCr}} = \begin{pmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{pmatrix}, \tag{3}$$

and

$$\mathbf{T}'_{\mathrm{RGB-to-YUV}} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{pmatrix}. \tag{4}$$

The Y component of the resultant image is usually called the luminance component, carrying most of the information, while the Cb and Cr components, or U V components, respectively, are called the chrominance components.

In our approach, we manipulate the luminance component, while leaving the chrominance components as they are, i.e., only the coefficients in the first row of the **T**-matrix are modified. The second and third rows of the

Table 1 The condition numbers of the baseline transforms and the mean and standard deviations of the condition number of our transforms for the Kodak dataset.

Baseline color transform	Condition number baseline	Condition number our transform
YCbCr	1.75	1.41 ± 0.07
Y′CbCr	1.75	1.38 ± 0.08
YUV	2.00	1.72 ± 0.05

matrix remain unaltered because in natural images, almost all of the image's energy is concentrated in the Y component.[11] As a result, most of the bits are allocated to the Y component. Consider this: The image "01," from the Kodak dataset[12] used in our experiments is coded with 2.03 bpp using standard JPEG with a quality factor of 80%. The PSNR is 33.39 dB. The Y component is coded with 1.76 bpp, while the chrominance components are coded with 0.27 bpp. Similarily, the "Barbara" image from our expanded dataset is coded with 1.69 bpp and a PSNR of 32.98 dB, when coded with a quality factor of 80%. The Y component is coded with 1.38 bpp, while the chrominance components are coded with 0.31 bpp.

Recent methods of color transform design include Refs. 13–15, but all of these methods try to optimize their designs over the entire image. However, different parts of a typical natural image may have different color characteristics. To overcome this problem, a block adaptive method taking advantage of the local color features of an image is proposed. In each block of the image, coefficients of the color transform are determined from the previously compressed neighboring blocks using weighted sums of the RGB pixel values, making the transform specific to that particular block.

We calculate the coefficients t_{11}, t_{12}, t_{13} of the first row of the color transform matrix, using the color content of the previous blocks in the following manner:

$$t_{11} = \frac{1}{2} \cdot \left[t'_{11} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i, j, 1)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k, l, m)} \right], \tag{5}$$

$$t_{12} = \frac{1}{2} \cdot \left[t'_{12} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i, j, 2)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k, l, m)} \right] \tag{6}$$

and

$$t_{13} = \frac{1}{2} \cdot \left[t'_{13} + \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} \mathbf{I}(i, j, 3)}{\sum_{k=1}^{M} \sum_{l=1}^{N} \sum_{m=1}^{3} \mathbf{I}(k, l, m)} \right], \tag{7}$$

where $\mathbf{I}$ denotes a three-dimensional, discrete RGB image composed of the used subimage blocks, which are to be discussed below, M and N denote the number of rows and columns of the subimage block, respectively, and t'_{1j} denotes the element in the 1st row and the j'th column of the 3×3 baseline color transform matrix, e.g., RGB-to-YCbCr. Normally, M and N are equal to 8 if only the previous block is used in JPEG coding.

Equations (5)–(7) have to be computed for each image block, therefore, the proposed transform changes for each block of the image. The extra overhead of encoding the color transform matrix can be easily avoided by borrowing an idea from standard differential pulse-code modulation (DPCM) coding in which predictor coefficients are estimated from encoded signal samples. In other words, there is no need to transmit or store the transform coefficients because they are estimated from previously encoded blocks. However, the specific 3×3 color transform matrix for a given block has to be inverted at the decoder. The computational cost for the inversion of a $N\times N$ matrix is usually given as $\mathcal{O}(N^3)$, however this is valid only in an asymptotic sense. For 3×3 matrices, a closed form expression exists, where the inverse can be found using 36 multiplications and 12 additions. In our case where we only alter the first row of the color transform matrix, this narrows down to 24 multiplications and 6 additions.

Since the color transform matrix is data specific, one may ask how numerically well-conditioned it is. A common technique to measure this is the condition number of a matrix. The condition number is defined as the ratio of the largest to the smallest singular value of the singular value decomposition of a given matrix.[16] A condition number with a value close to 1 indicates a numerically stable behavior of the matrix, i.e., it has full rank and is invertible. In order to investigate this, the condition number for each transform matrix of each block of the Kodak dataset was computed. Those results are averaged

Table 2 Mean and standard deviations of the correlation coefficients ρ_{ij} for the baseline color transforms and our transforms as computed over the Kodak dataset.

Color transform	ρ_{12}	ρ_{13}	ρ_{23}
YCbCr	-0.0008 ± 0.2250	-0.0436 ± 0.1308	-0.1683 ± 0.2372
Y′CbCr	-0.0006 ± 0.2260	-0.0444 ± 0.1306	-0.1691 ± 0.2369
YUV	-0.0008 ± 0.2264	-0.0444 ± 0.1308	-0.1696 ± 0.2378
Our YCbCr	0.0087 ± 0.2255	0.0043 ± 0.1308	-0.1683 ± 0.2337
Our Y′CbCr	0.0096 ± 0.2263	0.0073 ± 0.1306	-0.1691 ± 0.2337
Our YUV	0.0087 ± 0.2269	0.0041 ± 0.1308	-0.1696 ± 0.2343

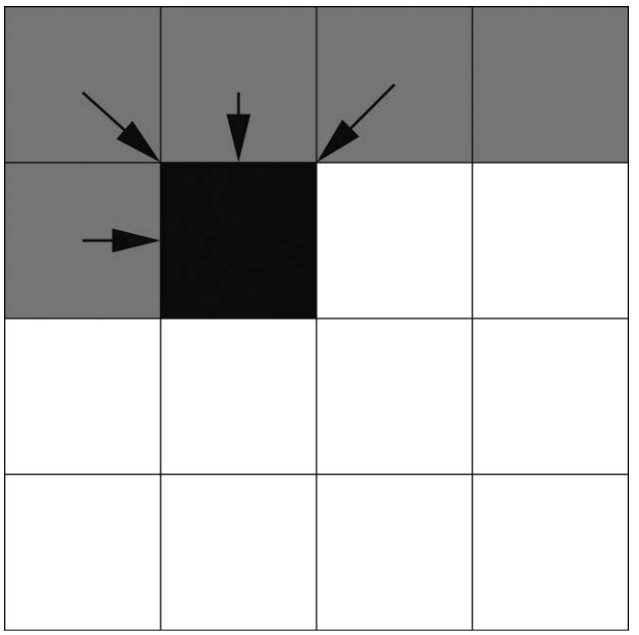
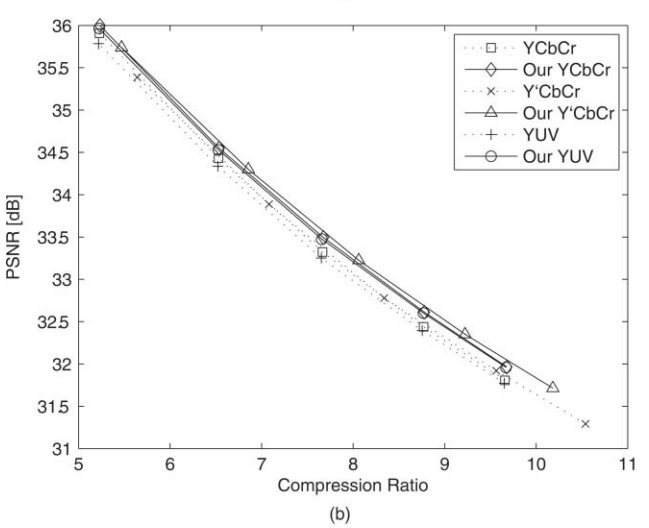


Fig. 1 A general description of our prediction scheme. To predict the color content of the black-shaded image block, color contents of previously encoded gray-shaded blocks, marked by arrows, are used.

(a)



(b)

Fig. 2 PSNR versus CR performance of the 24 image from the Kodak dataset for fixed color transforms and our method. (a) Original image, and (b) rate-distortion curve. Our method outperforms the baseline transforms.

Table 3 PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged. α is equal to 2.5.

Image	Average PSNR gain [dB] using YCbCr as baseline	Average PSNR gain [dB] using Y′CbCr as baseline	Average PSNR gain [dB] using YUV as baseline
1	0.0624	0.0928	0.0732
2	− 0.0668	− 0.0845	− 0.0368
3	− 0.2394	0.0824	− 0.6358
4	− 0.0325	0.0017	− 0.2449
5	0.0423	0.1008	0.0863
6	0.0966	0.1302	0.1564
7	0.0480	0.0767	0.0115
8	0.0958	0.1237	0.1326
9	0.1147	0.1349	0.1868
10	0.1395	0.2309	0.2167
11	0.0300	0.0791	0.0263
12	0.0781	0.0534	0.1261
13	0.1179	0.1162	0.1236
14	− 0.0517	− 0.0901	− 0.0538
15	− 0.0518	− 0.0486	0.0024
16	0.0812	0.1545	0.1415
17	0.0845	0.1265	0.1553
18	0.0952	0.1283	0.1113
19	0.0500	0.1003	0.0947
20	0.0399	0.0581	0.1320
21	0.0799	0.1535	0.1305
22	0.0642	0.1371	0.0762
23	− 0.4293	0.0525	− 1.1956
24	0.1448	0.1903	0.2081
1pmw	0.1832	0.1659	0.2331
ATI	0.0289	0.1388	0.1586
Airplane	0.5197	0.5079	0.4287
Baboon	0.0003	0.2097	− 0.4955
Barbara	0.1054	0.1294	0.1155
Boats	0.0913	0.0840	0.1348
DCTA	0.2134	0.2063	0.2457
Gl_Boggs	0.4427	0.3519	0.4673

Table 3 (*Continued.*)

Image	Average PSNR gain [dB] using YCbCr as baseline	Average PSNR gain [dB] using Y′CbCr as baseline	Average PSNR gain [dB] using YUV as baseline
Goldhill	0.2324	0.2395	0.2485
Huvahendhoo	0.2076	0.2254	0.2698
LagoonVilla	0.0791	0.0551	0.1229
Lake June	0.1223	0.1211	0.1388
Lenna	0.2070	0.2472	0.2197
Patrick	0.1130	0.0778	0.1489
Pepper	0.2158	0.2130	0.1769
PMW	0.1696	0.2188	0.2078
Serous-02	0.1245	0.0917	0.1606
Sunset Water Suite'	0.2036	0.3482	0.7866
Whole dataset	0.0967	0.1435	0.0952
Success rate	36/42	39/42	35/42

and can be seen alongside the values of the baseline transform matrices in Table 1. We find that for the given dataset, the condition number of our transform is in fact lower than the respective condition number of the baseline transform. It may also be of interest if our modified transform increases the interchannel correlation. In order to investigate this, the correlation coefficients ρ_{ij}, denoting the correlation between the i'th and j'th channel of a color transformed image, were calculated for the baseline transform matrices and for the modified transform matrices over the whole Kodak dataset. The average results can be seen in Table 2. We find that for the given dataset, the correlation between channels was not significantly increased by our method.

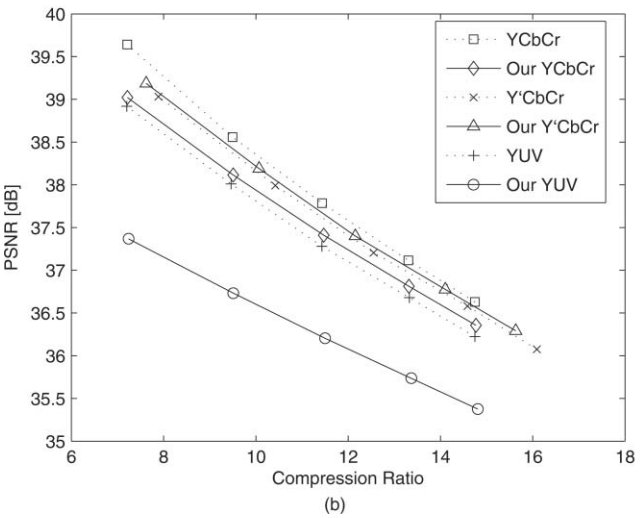
In most cameras, image blocks are raster-scanned from the sensor and blocks are fed to a JPEG encoder one by one.[5] For the first block of the image, the baseline color transform is used and the right-hand side of Eqs. (5)–(7) are computed from encoded–decoded color pixel values. For the second image block these color transform coefficients are inserted into the first row of the baseline color transform and it is encoded. The color content of the second block is also computed from encoded–decoded pixel values and used in the coding of the third block. Due to the raster-scanning, the correlation between neighboring blocks is expected to be high, therefore, for a given image block, the color content of its neighboring blocks is assumed to be a good estimate of its own color content. Furthermore, we are not restricted to use a single block to estimate the color transform parameters. We can also use image blocks of previously encoded upper rows as shown in Fig. 1 in which shaded blocks represent previously encoded blocks and the black shaded block is the current block. The neighboring blocks marked by an arrow are used for the prediction of the current block. In Ref. 17 an adaptive scheme is presented in which the encoder selects for each block of the image between the RGB, YCoCg, and a simple green, red-difference and blue-difference color spaces. This decision is signaled to the decoder as side information. Our method, however, does not require any transmission of side information to the receiver.

The current block's color content may be significantly different from previously scanned blocks. In such blocks we simply use the baseline color transform. Such a situation may occur if the current block includes an edge. We determine these blocks by comparing the color content with a threshold, as follows

$$\frac{1}{2} \cdot ||\mathbf{x}_c - \overline{\mathbf{x}}_p||_1 > \delta, \tag{8}$$

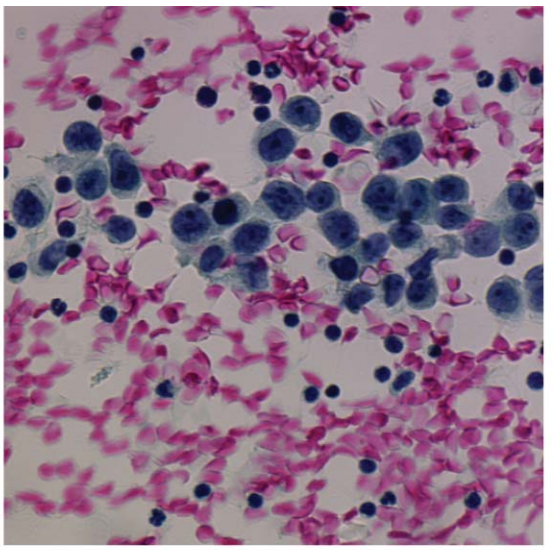
where $\mathbf{x}_c$ is the normalized weight vector of the current block's chrominance channels, $\overline{\mathbf{x}}_p$ is the mean vector of the chrominance channels' weights for all the neighboring blocks used in the prediction, and δ is the similarity threshold. The L1 norm was chosen due to its low computational cost. Note that in our prediction scheme we are not changing the chrominance channels. Therefore, we can use these for estimating the color content of the previous and current blocks,

(a)

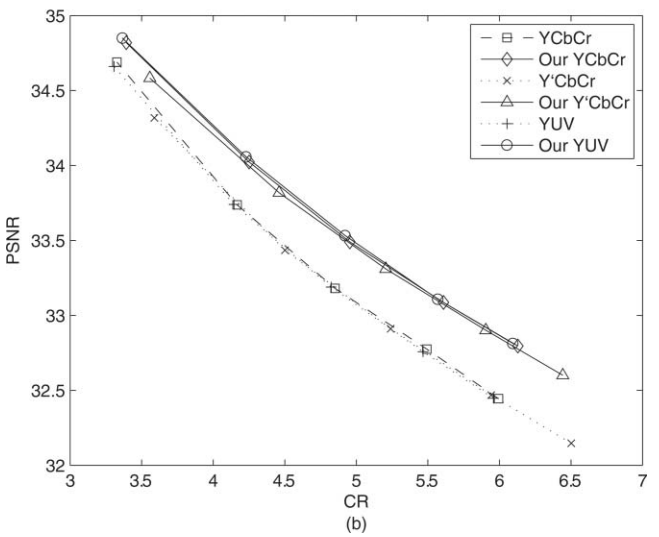


(b)

Fig. 3 PSNR versus CR performance of the 23 image from the Kodak dataset for fixed color transforms and our method. (a) Original image, and (b) rate-distortion curve. The baseline transforms outperform our method.



(a)



(b)

Fig. 4 PSNR versus CR performance of a microscopic image image from the Serous dataset for fixed color transforms and our method. (a) Original image, and (b) rate-distortion curve. Our method outperforms the baseline transforms.

regardless of the changes we make in the luminance channel. The threshold is chosen after investigating the values of the left-hand side of Eq. (8) for the Kodak dataset and calculating its mean and standard deviation. δ is then chosen according to

$$\delta = \mu + \alpha \cdot \sigma, \tag{9}$$

where μ and σ denote the mean and standard deviation of the left-hand side of Eq. (8), respectively α can take values between 2 and 3, since we assume a Gaussian model for the left-hand side of Eq. (8). In a Gaussian distribution, about 95.4% of the values are within two standard deviations around the mean ($\mu \pm 2 \cdot \sigma$), and about 99.7% of the values lie within 3 standard deviations around the mean ($\mu \pm 3 \cdot \sigma$). Therefore, in Eq. (8) we intend to measure if the L1 norm of the difference between the weight vectors of the current and the previous block lies within an interval of 2 to 3 standard deviations of the mean value. If it does not, we assume that

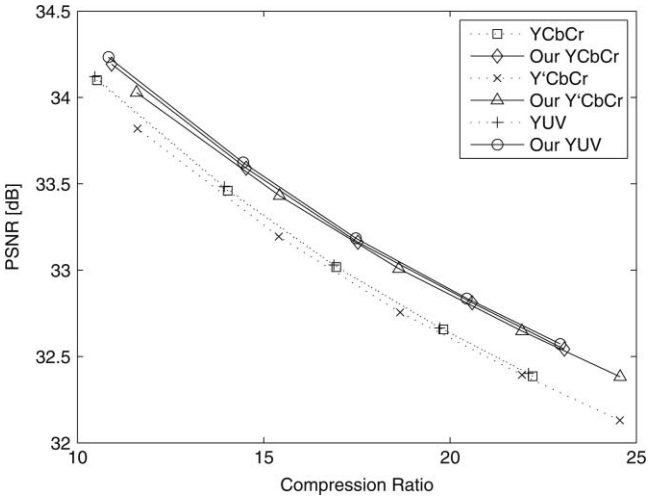


Fig. 5 PSNR versus CR performance of the Lenna image for fixed color transforms and our method. Rate-distortion curve. Our method outperforms the baseline transforms.

it is an outlier and therefore use the baseline color transform. In Sec. 3 we investigate the performance of several α values.

Due to our prediction scheme, no additional information on the color transform needs to be encoded by implementing a decoder inside the encoder as in standard DPCM signal encoding. It should also be pointed out that optimized color transform designs of Refs. 13, 14, and 15 can also be used in our DPCM-like coding strategy. Instead of estimating the color transform over the entire image the transform coefficients can be determined in the previously processed blocks as described above. The goal of this article is to introduce the block-adaptive color transform concept within the framework of JPEG and MPEG family of video coding standards. Therefore, a heuristic and a computationally simple color transform design approach is proposed in Eqs. (5)–(7). Since only the first row of the transform is modified it is possible to use the binary encoding schemes of JPEG and MPEG coders.

3 Experimental Studies and Results

A dataset of 42 images was used in our experiments. This includes the Kodak dataset, 10 high-resolution images ("1pmw," "ATI," "DCTA," "Gl_Boggs," "Huvahendhoo," "Patrick," "PMW," "LagoonVilla," "Lake June," "Sunset Water Suite"), the microscopic image "Serous-02" (Ref. 18) and the standard test images Lenna, Baboon, Goldhill, Boats,

Table 4 PSNR-gain values for the whole dataset with different baseline color transform. PSNR-gain of each image is measured at different rates and averaged. α is equal to 3.

Image	Average PSNR gain [dB] using YCbCr as baseline	Average PSNR gain [dB] using Y′CbCr as baseline	Average PSNR gain [dB] using YUV as baseline
Whole dataset	0.0948	0.1432	0.0941
Success rate	36/42	39/42	35/42

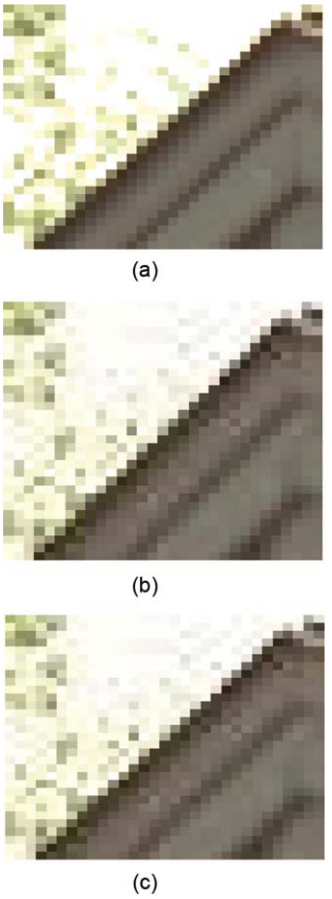


Fig. 6 A visual result of image 24 from the Kodak dataset coded by JPEG using a quality factor of 80%. (a) Original, (b) JPEG coded version using Y′CbCr, and (c) JPEG coded version using our method with Y′CbCr.

Pepper, Airplane, and Barbara. The high-resolution images have dimensions ranging from 1650×1458 to 2356×1579. The JPEG coder available in MATLAB's *imwrite* function is used in our experiments. The color transformation stage of the baseline JPEG is replaced with the proposed form of transformation. The weights of Eqs. (5)–(7) are computed using the previously processed blocks neighboring the current block as shown in Fig. 1.

We show several tables in which we alter the α value of Eq. (9). We choose α to be 2, 2.5, and 3, as explained in Sec. 2. The results can be seen in Tables 3–5. Results for using

Table 5 PSNR-gain values for the whole dataset with different baseline color transform. PSNR-gain of each image is measured at different rates and averaged. α is equal to 2.

Image	Average PSNR gain [dB] using YCbCr as baseline	Average PSNR gain [dB] using Y′CbCr as baseline	Average PSNR gain [dB] using YUV as baseline
Whole dataset	0.0967	0.1411	0.0965
Success rate	35/42	39/42	35/42

Table 6 PSNR-gain values for the whole dataset with different baseline color transform. PSNR-gain of each image is measured at different rates and averaged. No threshold was used, i.e., the whole image was coded with our method.

Image	Average PSNR gain [dB] using YCbCr as baseline	Average PSNR gain [dB] using Y′CbCr as baseline	Average PSNR gain [dB] using YUV as baseline
Whole dataset	0.0609	0.1207	0.0521
Success rate	31/42	34/42	31/42

no threshold at all, i.e., the whole image being coded by our method, can be seen in Table 6. Note that the δ threshold from Eq. (8) was computed using the data from the Kodak dataset but still performs well on the 18 additional images.

The PSNR-gain of our method over the baseline color transform is measured at five different compression ratios (CRs), 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 average gain value of a given image is positive. These results show that, on average, the proposed method produces better results than the baseline JPEG algorithm using the RGB-to-YCbCr, RGB-to-YUV, or RGB-to-Y′CbCr matrices, respectively. Using the threshold from Eq. (8) yields better results than using no theshold. On average, our method used with Y′CbCr yields the best compression gain. This tendency can be seen in most images. The two images were the performance of our method is the worst are images "3" and "23." In those cases, Y′CbCr has a positive gain but significantly smaller than its average gain, thus a negative tendency on the coding performance in those images is perceivable for all color spaces. Especially in those images, many sharp edges are visible and the color content on one side of the edge is not highly correlated to the color content on the other side of the edge. The differences in color content of certain blocks compared to their previous blocks are not as well detected by the threshold approach of Sec. 2 as in most other images of our dataset. The color estimate is therefore not accurate, resulting in a larger coding error.

In Figs. 2–5 the rate-distortion curves for 24, 23, Serous-02, and Lenna are given. While 24, Serous 2, and Lenna are images where our method outperforms the baseline transforms, in image 23 this is not the case. Images with strong, saturated color content that changes abruptly seem to perform worse with our method than with the baseline transform.

In Fig. 6, a visual example of our coding results is given. In Fig. 6(a), the original cropped image is shown, while Figs. 6(b) and 6(c) show the coded versions using Y′CbCR and our method based on Y'CbCr, respectively.

4 Conclusion

A method of extracting an image-specific color transform based on the color content of an image is presented. The transform coefficients are adaptively computed for each image block. The first row of the transform matrix is

determined by the color component ratios of previously compressed image blocks. 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. An application where the suggested method may be especially useful may be microscopic images where the color bandwidth is limited due to the staining process.

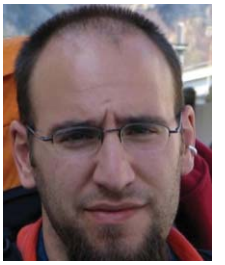
Acknowledgments

This study was funded by the Seventh Framework program of the European Union under grant agreement number PIRSES-GA-2009-247091 "MIRACLE-Microscopic Image Processing, Analysis, Classification and Modelling Environment."

References

1. G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.* **38**(1), 18–34 (1992).
2. E. Gershikov, E. Lavi-Burlak, and M. Porat, "Correlation-based approach to color image compression," *Signal Process. Image Commun.* **22**(9), 719–733 (2007).
3. Y. Chen, P. Hao, and A. Dang, "Optimal transform in perceptually uniform color space and its applications in image coding," *ICIAR 2004*, LNCS 3211, 269–276 (2004).
4. N. X. Lian, V. Zagorodnov, and Y. P. Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Trans. Image Process.* **5**(9), 2575–2587 (2006).
5. H. I. Cuce, A. E. Cetin, and M. K. Davey, "Compression of images in CFA format," *Proceedings of ICIP, IEEE*, pp. 1141–1144, Atlanta, GA (2006).
6. O. N. Gerek and A. E. Cetin, "Adaptive polyphase subband decomposition structures for image compression," *IEEE Trans. Image Process.* **9**(10), 1649–1660 (2000).
7. O. N. Gerek and A. E. Cetin, "A 2-D orientation-adaptive prediction filter in lifting structures for image coding," *IEEE Trans. Image Process.* **15**(1), 106–111 (2006).
8. H. J. Trussell and M. J. Vrhel, "Foundations of digital imaging," Cambridge Press, Cambridge, UK (2008).
9. G. Sharma and H. J Trussell, "Digital color processing," *IEEE Trans. Image Process.* **IP–6**(7), 901–932 (1997).
10. E. Hamilton, "JPEG file interchange format (Version 1.02)," C-Cube Microsystems, Milpitas, California (1992).
11. N. S. Jayant and P. Noll, "Digital coding of waveforms: Principles and applications to speech and video," Prentice-Hall, Englewood Cliffs, NJ (1990).
12. http://r0k.us/graphics/kodak/
13. E. Gershikov and M. Porat, "On color transforms and bit allocation for optimal subband image compression," *Signal Process. Image Commun.* **22**(1), 1–18 (2007).
14. Y. Ohta, T. Kanade, and T. Sakai, "Color information for region segmentation," *Comput. Graph. Image Process.* **13**(3), 222–241 (1980).
15. W. K. Pratt, "Spatial transform coding of color images," *IEEE. Trans. Commun.* **19**, 980–992 (1971).
16. T. K. Moon and W. C. Stirling, "Mathematical methods and algorithms for signal processing," Prentice-Hall, Englewood Cliffs, NJ (2000).
17. D. Marpe, H. Kirchhoffer, V. George, P. Kauff, and T. Wiegand "An adaptive color transform approach and its application in 4:4:4 video coding," *Proc. 14th European Signal Processing Conference (EUSIPCO 2006)*, Florence, Italy, September 4–8, 2006.
18. O. Lezoray and H. Cardot, "Cooperation of color pixel classification schemes and color watershed: a study for microscopical images," *IEEE Trans. Image Process.* **11**(7), 783–789 (2002).

Alexander Suhre received his Diplom-Ingenieur degree in electrical engineering and information technology from TU Darmstadt, Germany in 2006. In 2005 he spent 6 months as a visiting researcher at the Centre for Eye Research at the Queensland University of Technology, Brisbane, Australia. He is currently working toward his PhD in electrical engineering at Bilkent University, Ankara, Turkey. His research interests include signal and image processing, computer vision, as well as image compression.



Kivanc Kose is currently working through his PhD degree at the Electrical and Electronics Engineering Department at Bilkent. He studied the compression of the 3D mesh models during his MSc period under the supervision of Professor Enis Cetin. He implemented a new orthographic projection method for 3D model. Moreover, he implemented a new adaptive wavelet transformation called connectivity guided adaptive wavelet transformation, for this projected 2D model. He has experience on adaptive wavelet transformation and its applications in image processing. He has published 10 conference and 1 journal paper.

A. Enis Cetin received his MSE and PhD degrees from the University of Pennsylvania, Pennsylvania. Between 1987 and 1989, he was assistant professor at the University of Toronto, Canada. Since then he has been with Bilkent University, Ankara, Turkey. In 1996, he was promoted to the position of professor. During the summers of 1988, 1991, and 1992 he was with Bell Communications Research (Bellcore) as a consultant. He spent the 1996 to 1997 academic year at the University of Minnesota, Minneapolis, Minnesota as a visiting professor. He carried out contract research for both governmental agencies and industry including Visioprime, Surrey, UK; Honeywell Video Systems; National Science Foundation–USA, NSERC–Canada; ASELSAN, Ankara, Turkey. He is one of the founders of the camera design company Grandeye, UK. He was a scientific committee member of the FP-6 Network of Excellence (NoE) MUSCLE: Multimedia Understanding through Semantics, Computation and Learning during 2004–2008. He also took part in the FP-6 project: 3DTV. Between 1999 and 2003, he was an associate editor of the IEEE Transactions on Image Processing. Currently, he is on the editorial boards of Signal Processing, EURASIP Journal of Advances in Signal Processing and Journal of Machine Vision and Applications. He is a member of the DSP technical committee of the IEEE CAS Society. He was the co-chair of the IEEE-EURASIP Nonlinear Signal and Image Processing Workshop (NSIP'99) and was the technical co-chair of the European Signal Processing Conference-EUSIPCO-2005. He received the young scientist award of TUBITAK (Turkish Scientific and Technical Research Council) in 1993.

Metin N. Gurcan received his BSc and PhD degrees in electrical and electronics engineering from Bilkent University, Turkey and his MSc degree in digital systems engineering from the University of Manchester Institute of Science and Technology, England. He is the recipient of the British Foreign and Commonwealth Organization Award, Childrens Neuroblastoma Cancer Foundation Young Investigator Award, and National Cancer Institutes caBIG Embodying the Vision Award. During the winters of 1996 and 1997 he was a visiting researcher at the University of Minnesota, Minneapolis. From 1999 to 2001, he was a postdoctoral research fellow and later a research investigator in the Department of Radiology at the University of Michigan, Ann Arbor, Michigan. Prior to joining the Ohio State University in October 2005, he worked as a senior researcher and product director at a high-tech company, specializing in computer-aided detection and diagnosis of cancer from radiological images. He is a senior member of IEEE, SPIE, and RSNA.