

# Content-Adaptive Color Transform For Image Compression

Alexander Suhre, Kivanc Kose, and A. Enis Cetin

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

Metin N. Gurcan

*Ohio State University, Department of Biomedical Informatics, Columbus, OH 43210, USA<sup>†</sup>*

(Dated: January 4, 2011)

## 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 PSNR values for a given compression level.

## I. INTRODUCTION

Image compression is a well-established and extensively studied field in the signal processing and communication communities. Although the “lossy” JPEG standard<sup>1</sup> is one of the most widely accepted image compression technique in modern day applications, its resulting fidelity can be improved. 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<sup>2</sup>. An optimal solution would be to use Karhunen-Loève Transform (KLT), see<sup>3</sup>. However, in KLT there is an underlying wide-sense stationary random process assumption which may not be valid in natural images. Another approach to an optimal color space projection on a four-dimensional colorspace was developed in<sup>4</sup>.

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<sup>5, 6</sup> and<sup>7</sup>.

## II. ALGORITHM

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

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

where  $\mathbf{T} = [t_{ij}]_{3 \times 3}$  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, see<sup>8</sup> and<sup>7</sup>. For example, JPEG uses luminance-chrominance type colorspace transforms and chooses the coefficients in  $\mathbf{T}$  accordingly. Examples for these include JFIF’s RGB-to-YCbCr<sup>9</sup>, 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 as

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

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

and

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

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  $\mathbf{T}$ -matrix are modified. The second and third rows of the matrix remain unaltered because in natural images, almost all of the image's energy is concentrated in the Y component<sup>10</sup>. As a result, most of the bits are allocated to the Y component. Consider this: The image '01', from the 'Kodak' dataset<sup>11</sup> used in our experiments, is coded with 2.03 bits per pixel (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. Similarly, 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<sup>12, 13</sup> and<sup>14</sup>, 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:

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

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

and

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

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 1-st row and the  $j$ -th column of the 3-by-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 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-by-3 color transform matrix for a given block has to be inverted at the decoder.

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 relation of the largest to the smallest eigenvalue of a given matrix<sup>16</sup>. A condition number with value close to 1 indicates a numerically stable behaviour 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 and can be seen alongside the values of the baseline transform matrices in Table I. 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  $\rho_{ij}$ , denoting the correlation between

Baseline	Condition	Condition
Color	Number	Number
Transform	Baseline	Our Transform
YCbCr	1.75	1.41
Y'CbCr	1.75	1.38
YUV	2.00	1.72

TABLE I: The condition numbers of the baseline transforms and the mean of the condition number of our transforms for the 'Kodak' dataset.

Color	$\rho_{12}$	$\rho_{13}$	$\rho_{23}$
Transform			
YCbCr	-0.0008	-0.0481	0.1683
Y'CbCr	-0.0006	-0.0488	0.1691
YUV	-0.0008	-0.0484	0.1696
Our YCbCr	0.0087	-0.0080	0.1683
Our Y'CbCr	0.0096	-0.0056	0.1691
Our YUV	0.0087	-0.0083	0.1696

TABLE II: The average correlation coefficients  $\rho_{ij}$  for the baseline color transforms and our transforms as computed over the 'Kodak' dataset.

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 mean results can be seen in Table II. 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<sup>5</sup>. For the first block of the image, the baseline color transform is used and the right-hand side of Equations (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 Figure 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<sup>15</sup> 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

$$\frac{1}{2} \cdot \|\mathbf{x}_c - \bar{\mathbf{x}}_p\|_1 > \delta, \quad (8)$$

where  $\mathbf{x}_c$  is the normalised weight vector of the current block’s chrominance channels,  $\bar{\mathbf{x}}_p$  is the mean vector of the chrominance channels’ weights for all the neighboring blocks used in the prediction and  $\delta$  is the similarity threshold. 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, regardless of the changes we make in the luminance channel. The threshold is chosen after investigating the values of the left hand side of Equation 8 for the ‘Kodak’ dataset and calculating its mean and standard deviation.  $\delta$  is then chosen according to

$$\delta = \mu + \alpha \cdot \sigma. \quad (9)$$

where  $\mu$  and  $\sigma$  denote the mean and standard deviation of the left hand side of Equation 8, respectively.  $\alpha$  can take values between 2 and 3, since we assume a Gaussian model for the left hand side of Equation 8. In a Gaussian distribution, 95% 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 ( $\mu \pm 3 \cdot \sigma$ ). In Section III we investigate the performance of several  $\alpha$  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 be also pointed out that optimized color transform designs of<sup>12,13</sup> and<sup>14</sup> can be also 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 Equations (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.

### III. EXPERIMENTAL STUDIES AND RESULTS

A dataset of 41 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') and the standard test images Lenna, Baboon, Goldhill, Boats, Pepper, Airplane and Barbara. The high-resolution images have dimensions ranging from 1650-by-1458 to 2356-by-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 Equations (5)-(7) are computed using the previously processed blocks neighboring the current block as shown in Figure 1.

We show several tables, in which we alter the  $\alpha$  value of Equation 9. We choose  $\alpha$  to be 2, 2.5 and 3, as explained in Section II. The results can be seen in Tables III-V. Results for using no threshold at all, i.e., the whole image being coded by our method, can be seen in Table VI. Note that the  $\delta$  threshold from Equation 8 was computed using the data from the 'Kodak' dataset but still performs well on the 14 additional images.

The PSNR-Gain of our method over the baseline color transform is measured at five different compression ratios (CR), 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.

In Figures IV - IV the rate-distortion curves for '24', '23' and 'Lenna' are given. While '24' 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 Figure IV, a visual example of our coding results is given. In Figure IV (a), the original cropped image is shown, while Figures IV (b)-(c) show the coded versions using Y'CbCR and our method based on Y'CbCr, respectively.

#### IV. 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.

- 
- \* Electronic address: [suhre@ee.bilkent.edu.tr](mailto:suhre@ee.bilkent.edu.tr)
- † Electronic address: [metin.gurcan@osumc.edu](mailto:metin.gurcan@osumc.edu)
- <sup>1</sup> G.K. Wallace, “The JPEG Still Picture Compression Standard”, *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 1 pp. 18-34, 1992.
- <sup>2</sup> E. Gershikov, E. Lavi-Burlak and M. Porat, Correlation-based approach to color image compression, *Signal Processing: Image Communication*, Volume 22, Issue 9 pp. 719-733, 2007.
- <sup>3</sup> Y. Chen, P. Hao and A. Dang, “Optimal Transform in Perceptually Uniform Color Space and Its Applications in Image Coding”, *ICIAR 2004, LNCS 3211* pp. 269276, 2004.
- <sup>4</sup> N.X. Lian, V. Zagorodnov, Y.P. Tan , “Edge-Preserving Image Denoising via Optimal Color Space Projection”, *IEEE Trans Image Process.*, Volume 5, Issue 9 pp. 2575-87, 2006.
- <sup>5</sup> H. I. Cuce, A. E. Cetin, M. K. Davey, “Compression of images in CFA format”, *Proc. of ICIP, 2006* pp. 1141-1144, 2006.
- <sup>6</sup> 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.
- <sup>7</sup> 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.
- <sup>8</sup> H. J. Trussell and M. J. Vrhel, “Foundations of Digital Imaging”, Cambridge Press, December 2008, copyright 2009.
- <sup>9</sup> E. Hamilton, “JPEG File Interchange Format (Version 1.02),” C-Cube Microsystems, Milpitas, California, 1992.
- <sup>10</sup> N.S. Jayant and P. Noll, “Digital Coding of Waveforms: Principles and Applications to Speech and Video”, Prentice-Hall, 1990.
- <sup>11</sup> <http://r0k.us/graphics/kodak/>
- <sup>12</sup> 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.
- <sup>13</sup> 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.
- <sup>14</sup> W. K. Pratt, “Spatial transform coding of color images”, *IEEE. Trans. Commun.*, vol. 19 pp.

980-992, 1971.

- <sup>15</sup> 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.
- <sup>16</sup> T.K. Moon, W.C. Stirling: “Mathematical Methods and Algorithms for Signal Processing”, Prentice-Hall, 2000.

## Tables

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
'ATF'	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
'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
'Sunset Water Suite'	0.2036	0.3482	0.7866
Whole dataset	0.0910	0.1376	0.0886
Success rate	35/41	38/41	34/41

TABLE III: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged.  $\alpha$  is equal to 2.5.

Image	Average PSNR	Average PSNR	Average PSNR
	Gain [dB]	Gain [dB]	Gain[dB]
	using YCbCr as baseline	using Y'CbCr as baseline	using YUV as baseline
Whole dataset	0.0888	0.1370	0.0870
Success Rate	35/41	38/41	34/41

TABLE IV: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged.  $\alpha$  is equal to 3.

Image	Average PSNR	Average PSNR	Average PSNR
	Gain [dB]	Gain [dB]	Gain[dB]
	using YCbCr as baseline	using Y'CbCr as baseline	using YUV as baseline
Whole dataset	0.0913	0.1355	0.0905
Success Rate	34/41	38/41	34/41

TABLE V: PSNR-Gain values for the whole dataset with different baseline color transform. PSNR-Gain of each image is measured at different rates and averaged.  $\alpha$  is equal to 2.

Image	Average PSNR	Average PSNR	Average PSNR
	Gain [dB]	Gain [dB]	Gain[dB]
	using YCbCr as baseline	using Y'CbCr as baseline	using YUV as baseline
Whole dataset	0.0535	0.1133	0.0430
Success Rate	30/41	33/41	30/41

TABLE VI: 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.

## Figures

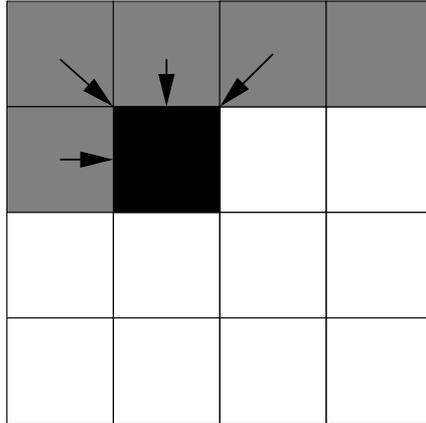
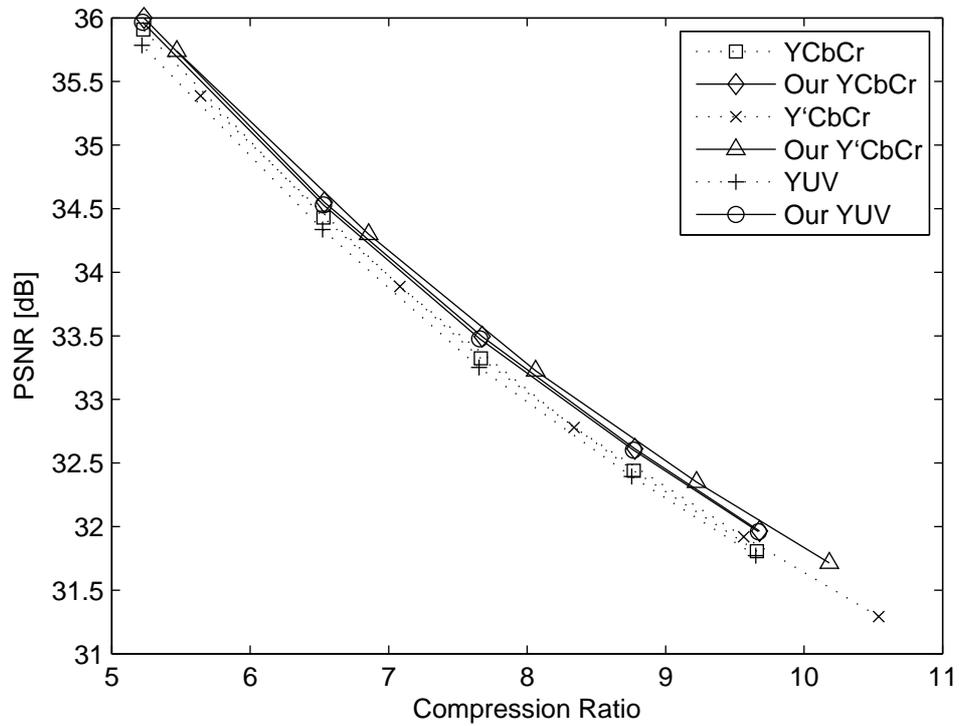


FIG. 1: A general description of our prediction scheme. To predict the color content of the black-shaded image block, color contents of previously encoded gray-shaded blocks, marked by arrows, are used.



(a)

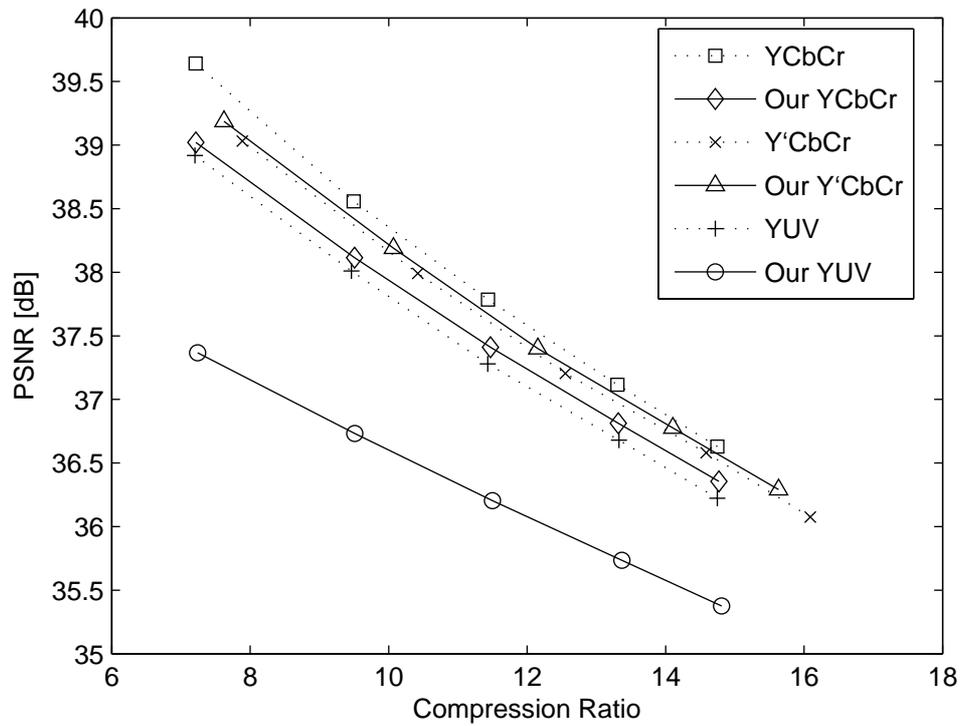


(b)

FIG. 2: PSNR-vs-CR performance of the '24' image from the 'Kodak' dataset for fixed color transforms and our method. (a) Original Image, (b) Rate-Distortion curve. Our method outperforms the baseline transforms.



(a)



(b)

FIG. 3: PSNR-vs-CR performance of the '23' image from the 'Kodak' dataset for fixed color transforms and our method. (a) Original Image, (b) Rate-Distortion curve. The baseline transforms outperform our method.

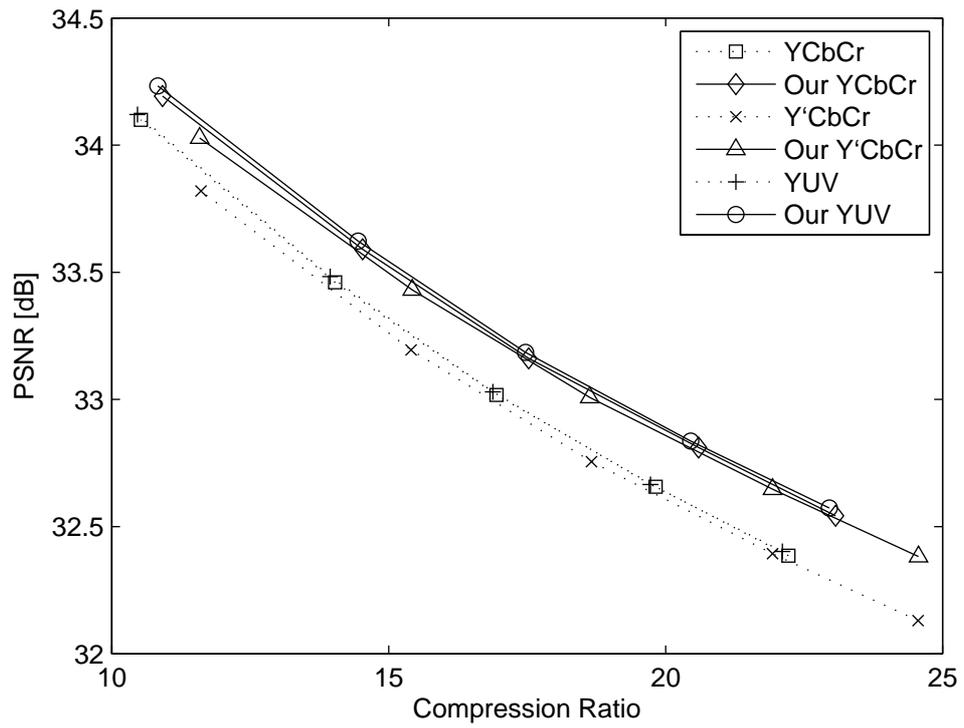
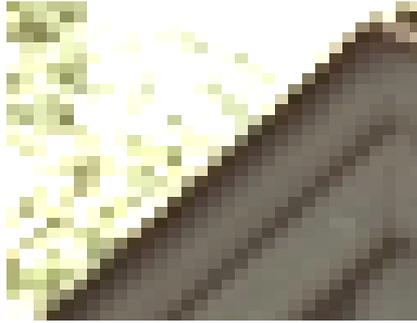


FIG. 4: PSNR-vs-CR performance of the 'Lenna' image for fixed color transforms and our method. Rate-Distortion curve. Our method outperforms the baseline transforms.



(a)



(b)



(c)

FIG. 5: 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.