COLOR IMAGE CODING BASED ON LINEAR COMBINATION OF ADAPTIVE COLORSPACES

Takanori Fujisawa, Masaaki Ikehara

EEE Dept., Keio Univ., Yokohama, Kanagawa 223-8522, Japan
{fujisawa,ikehara}@tkhm.elec.keio.ac.jp

ABSTRACT

This paper improves a colorization-based image coding using image segmentation and adaptive colorspaces. Recently, various approaches for color image coding based on colorization have been presented. These methods utilize a YCbCr colorspace and transfer the luminance component by a conventional compression method. Then, the chrominance components are approximated from the luminance component using a colorization method. Our method segments a luminance component into small segments called superpixels, and reconstructs the chrominance of each superpixel as a linear combination of its luminance. For chrominance components, we introduce an adaptive color space transform optimized for liner combination. This is because YCbCr colorspace cannot always become a good approximation of the chrominance. In addition, we introduce an automatic selection for the number of superpixel segments from a given quality factor. The simulation with standard images shows that our method performs better result than conventional coding schemes.

Index Terms — Image coding, Colorization, Inter-channel color cross-correlation, Color transform

1. INTRODUCTION

Color image is typically represented in a RGB color space. Many image coding methods convert the RGB color space to the YCbCr color space before the main process. For example, the international standard JPEG [1] coding scheme adopts this YCbCr color space and applies DCT to Y, Cb and Cr components separately. The components of Cb, Cr are commonly said to have less variance compared to the components in Y [2]. Therefore, most image coding methods reduce the Cb or Cr's information and achieve high compression ratio. This separative transform achieves fast computation, however it cannot be said that it effectively utilizes the correlation between the luminance and the chrominance components.

The colorization-based coding schemes improve the coding efficiency by utilizing the correlation between color components. Many colorization-based coding methods have been proposed in recent years [3–10]. These methods utilize a colorization method [11] to fill up the chrominance information. Their encoder selects a few representative pixels to transfer. The decoder reconstructs chrominance value considering the correlation to a known luminance values. To achieve an efficient coding, colorization-based coding needs to represent chrominance information with a small amount of bits and a few reconstruction errors.

We introduce an another approach for approximating the chrominance components. Our method segments an input image into small segments using SLIC algorithm [13]. Chrominance values are approximated in each segment by a linear combination of luminance values. An encoder of this method transfers coefficients of the linear approximation to decoder side. The decoder reconstructs chrominance values from luminance signals which was transferred by another method. We additionally introduce an optimal colorspace transform to the linear combination. This colorspace can reduce the approximation error and improve the quality of decoded output.

The rest of this paper is organized as follows. Section 2 describes the coding scheme based on colorization. We present the improvement of this method in Section 3. Experimental results are shown and discussed in Section 4. Finally, Section 5 concludes this paper.

2. COLORIZATION IMAGE CODING ON SUPERPIXEL SEGMENTS

Our method converts an RGB image into an adaptive colorspace to ensure the correlation between luminance and chrominance components. The luminance component is segmented into small segments using SLIC algorithm [13]. The chrominance values are approximated in each segment by a linear combination of luminance values. Our method transfers the luminance component by JPEG and the coefficients representing the chrominance components by a Huffman code. The decoder segments the luminance component transferred by the encoder. The chrominance components are reconstructed in each segment by the linear approximation using the transferred coefficients. An overall framework of our encoder, decoder and coding scheme of color image is shown in Fig. 1.

2.1. Image coding based on cross-layer correlation

Image coding frameworks such as JPEG utilize the color transform to RGB image to other layer, and each layer is encoded by different coding schemes. The transformed colorspace can be expressed using the 3×3 transform matrix to the pixel of RGB colorspace. YCbCr colorspace used by JPEG is expressed as the transform matrix shown as follows:

$$\begin{bmatrix} y \\ c_b \\ c_r \end{bmatrix} = \begin{bmatrix} 0.2571 & 0.5041 & 0.0979 \\ -0.1687 & -0.3313 & 0.5000 \\ 0.5000 & -0.4187 & -0.0813 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{1}$$

The layer shown as y pixels is called as a luminance layer and the layers of c_b, c_r are chrominance layers.

The idea of the colorization-based image coding assumes a correlation between the luminance layer and the chrominance layer in a local region. Our coding scheme approximates the chrominance pixel values as a linear combination of the luminance values as well as [10, 12].

$$\boldsymbol{c} \simeq a\boldsymbol{y} + b\boldsymbol{1} \tag{2}$$

In equation (2), the expression $\boldsymbol{c}$ denotes the vector of chrominance pixels in a local region. Similarly, $\boldsymbol{y}$ denotes the pixels of the luminance layer. The coefficients a and b indicate how chrominance

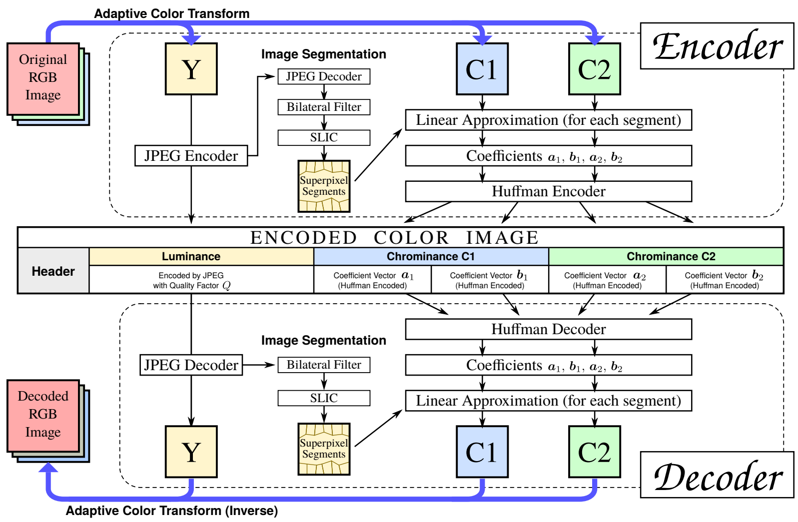


Fig. 1. An overview of encoder, decoder and an encoded color image of the proposed method.

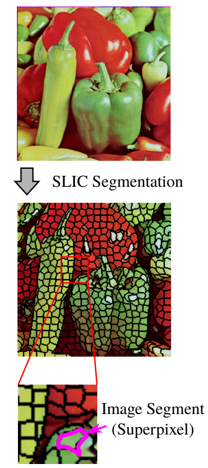


Fig. 2. Image segmentation using SLIC superpixels.

layer is approximated from the given luminance value. Therefore transmitting the pair (a, b) instead of chrominance vector $\boldsymbol{c}$ achieves the reduction of bit amount. In other words, it is suitable for encoding scheme for chrominance layer. The coefficients pair (a, b) is easily calculated by a least squares problem shown as follows. The superscript † denotes Moore-Penrose inverse operation.

$$\begin{bmatrix} a \\ b \end{bmatrix} = \operatorname*{argmin}_{a,b} \left\| \boldsymbol{c} - \begin{bmatrix} \boldsymbol{y} & \boldsymbol{1} \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} \right\|_2^2 \simeq \begin{bmatrix} \boldsymbol{y} & \boldsymbol{1} \end{bmatrix}^{\dagger} \boldsymbol{c} \quad (3)$$

2.2. Chrominance coding over SLIC superpixels

The luminance layer and the chrominance layer has less correlation when the region has many image features such as edges. To enforce the precision of linear approximation, our framework divides the luminance image into segments. The chrominance layer is encoded for each divided segments. For the image segmentation, Achanta's SLIC algorithm [13] is utilized. In this algorithm, a luminance image is segmented into regions shown in Fig. 2, which are called superpixels. SLIC takes a parameter K to indicate the number of superpixels. Larger K generates smaller sized superpixels, which reduce the approximation error. On the other hand, the larger number of superpixels makes more coefficients to transfer and increases the bits to express them. Before applying SLIC, we apply a bilateral filter for the luminance layer. The filter suppresses the texture components while preserving the edge components. This process reduces the approximation error of the chrominance layer.

2.3. The encoding scheme for chrominance layer

After linear approximations (3) for each superpixel i, we obtain sets of coefficients $\boldsymbol{a}_1 = [a_1, a_2, \ldots, a_K]$ and $\boldsymbol{b}_1 = [b_1, b_2, \ldots, b_K]$ for Cb colorspace, and $\boldsymbol{a}_2, \boldsymbol{b}_2$ for Cr colorspace. The elements of these sets have a small variance as shown in Fig. 3. Therefore, we utilize a Huffman code to express them in smaller amount of bits. Before encoding by Huffman code, we quantize coefficient values a, b to integers of range 0 to 255. In addition to the Huffman code, we need to transfer the number of occurrences of the coefficients to built a Huffman code table. Many of the occurrences have the value from 0 to 4, the small occurrences are encoded by small amount bits as shown in Fig. 4. Four coding schemes are prepared so that the encoder can select the most efficient one. The whole codes to explain the chrominance layer is shown in Fig. 5. Note that the pair of codes described in Fig. 5 appears 4 times in order to transfer the vector $\boldsymbol{a}_1, \boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{b}_2$ respectively.

3. COLOR IMAGE CODING WITH ADAPTIVE COLORSPACES

3.1. Designing an adaptive colorspace

The YCbCr colorspace is originally intended to suppress the variance of the chrominance layer. Therefore, the luminance layer cannot always approximate the chrominance layer correctly. In order to reduce the approximation error, we introduced an adaptive colorspace instead of YCbCr colorspace. As well as YCbCr color transform, our adaptive colorspace is expressed as a transform matrix to RGB value. However, the chrominance values c_b and c_r are replaced by "chrominance-like" values c_1 and c_2 expressed by the weights w_1 to w_6 in equation (4).

$$\begin{bmatrix} y \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0.2571 & 0.5041 & 0.0979 \\ w_1 & w_2 & w_3 \\ w_4 & w_5 & w_6 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \quad (4)$$

These weights w_1 to w_6 are defined adaptively to the images and transmitted to the decoder.

The decision of these weights is achieved by minimizing the approximation error. This error was caused from the linear combination of luminance layer. In other words, the weight vector $\boldsymbol{w} = [w_1, w_2, \ldots, w_6]$ can be obtained by the following optimization problem (5). The objective function $d(\boldsymbol{w})$ sums up the approximation error of all superpixels.

$$\boldsymbol{w} = \operatorname*{argmin}_{\boldsymbol{w}} d(\boldsymbol{w}); \; d(\boldsymbol{w}) = \sum_i^K \|\boldsymbol{c}_{1,i} - (a_{1i}\boldsymbol{y}_i + b_{1i}\boldsymbol{1})\|_2^2 + \sum_i^K \|\boldsymbol{c}_{2,i} - (a_{2i}\boldsymbol{y}_i + b_{2i}\boldsymbol{1})\|_2^2 \quad (5)$$

a_{1i} and b_{1i} represent coefficients of the i-th superpixel, which approximate the chrominance values $\boldsymbol{c}_{1i}$ from luminance values $\boldsymbol{y}_i$. Likewise, a_{2i} and b_{2i} represent the chrominance $\boldsymbol{c}_{2i}$ from $\boldsymbol{y}_i$. According to (3), the approximation error on C_1 layer can be formulated as follows.

$$\begin{aligned} &\sum_i^K \|\boldsymbol{c}_{1,i} - (a_{1i}\boldsymbol{y}_i + b_{1i}\mathbf{1})\|_2^2 \\ &= \sum_i^K \|\boldsymbol{c}_{1i} - \begin{bmatrix}\boldsymbol{y}_i & \mathbf{1}\end{bmatrix} \left(\begin{bmatrix}\boldsymbol{y}_i & \mathbf{1}\end{bmatrix}\right)^\dagger \boldsymbol{c}_{1i}\|_2^2 \\ &= \sum_i^K \|(\boldsymbol{I} - \boldsymbol{Y}_i)(w_1\boldsymbol{r}_i + w_2\boldsymbol{g}_i + w_3\boldsymbol{b}_i)\|_2^2 \end{aligned} \tag{6}$$

Here, we utilized an expression $\boldsymbol{I}$ as an identity matrix and $\boldsymbol{Y}_i$ as a square matrix of following equation.

$$\boldsymbol{Y}_i = \begin{bmatrix}\boldsymbol{y}_i & \mathbf{1}\end{bmatrix} \begin{bmatrix}\boldsymbol{y}_i & \mathbf{1}\end{bmatrix}^\dagger \tag{7}$$

On C_2 layer, the objective function has the same expression as (6) with weights w_4, w_5, w_6 instead of w_1, w_2, w_3. From (5) and (6), the objective function $d(\boldsymbol{w})$ can be rewritten as follows.

$$\begin{aligned} d(\boldsymbol{w}) = &\sum_i^K \|(\boldsymbol{I} - \boldsymbol{Y}_i)(w_1\boldsymbol{r}_i + w_2\boldsymbol{g}_i + w_3\boldsymbol{b}_i)\|_2^2 \\ &+ \sum_i^K \|(\boldsymbol{I} - \boldsymbol{Y}_i)(w_4\boldsymbol{r}_i + w_5\boldsymbol{g}_i + w_6\boldsymbol{b}_i)\|_2^2 \end{aligned} \tag{8}$$

The optimization of (8) is an ill-posed problem. That is, many pairs of w's satisfy the formulation. The value of w should be in the range of $(-2, 2)$ to encode with higher precision and to suppress arithmetic underflow when transforming $YC_1C_2 \rightarrow RGB$ colorspaces.

3.2. Learning the default color transform

The optimization of colorspaces requires many inverse problems to calculate $\boldsymbol{Y}_i$, which increases the execution time for the encoding process. Considering the efficiency, preparing the optimized color transform matrix in advance will be recommended.

The pre-defined transform matrix can be obtained by learning process with some test images. The formulation of the learning process from N_j images becomes as follows. $d_j(\boldsymbol{w})$ is a total approximation error of j-th test image.

$$\boldsymbol{w} = \operatorname*{argmin}_{\boldsymbol{w}} \sum_j^{N_j} d_j(\boldsymbol{w}) \tag{9}$$

For example, the color transform matrix learned from 15 standard images becomes as follows:

$$\begin{bmatrix} y \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0.2571 & 0.5041 & 0.0979 \\ 0.7039 & 0.1193 & -0.7472 \\ -0.8055 & 1.3331 & -0.9933 \end{bmatrix} \begin{bmatrix} r \\ g \\ b \end{bmatrix} \tag{10}$$

4. AUTOMATIC PARAMETER SELECTION OF SLIC

In the practical use, estimating the optimal number of superpixel segments K in the encoding is not always possible. To deal with this problem, we describe an automatic selection of the SLIC parameter K. The parameter K is decided from the quality factor Q, which is given for encoding the luminance layer by JPEG. We find out that

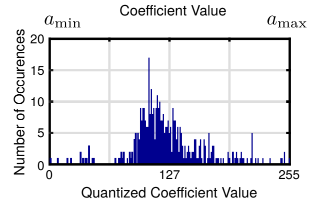


Fig. 3. Quantized coefficients of a linear approximation.

	Number of Occurences					
	0	1	2	3	4	5 and more
Scheme 1	`10`	`11`	`Flag 0 & Occurence (8bit)`			
Scheme 2	`10`	`110`	`111`	`Flag 0 & Occurence (8bit)`		
Scheme 3	`10`	`110`	`1110`	`1111`	`Flag 0 & Occurence (8bit)`	
Scheme 4	`10`	`110`	`1110`	`11110`	`11111`	`Flag 0 & Occurence (8bit)`

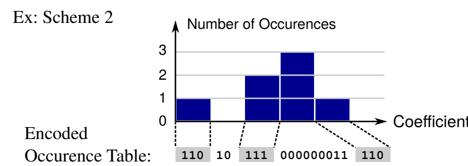


Fig. 4. The encoding scheme of a number of occurrences.

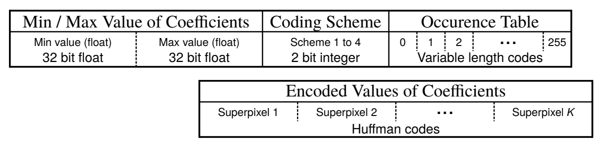


Fig. 5. A whole code of coefficient vector $\boldsymbol{a}_1$ ($\boldsymbol{b}_1, \boldsymbol{a}_2, \boldsymbol{b}_2$ as well.)

the relation between Q and an optimal K can be approximated by an exponential function. Therefore we formulated the relation as follows and obtained a optimal pair of coefficients (α, β, γ) for some standard images.

$$\begin{aligned} K &= \gamma \exp(\alpha Q + \beta) \\ \beta &= \left(0.0033u^{-1} + 0.0015\right) N \\ \alpha &= \frac{1}{90}(\ln 200 - \beta), \quad \gamma = 5 \end{aligned} \tag{11}$$

Here, N denotes the number of pixels in the image and u denotes the variance of the histogram of luminance pixels. The automatic parameter selection scheme reduces the execution time for segmentation process.

5. EXPERIMENTAL RESULTS

The evaluation is executed for 6 standard images of size 256×256 and 5 images of size 512×512. As a quantitative evaluation, PSNR metrics are calculated for the decoded output. The compression ratio is selected from 2.5% to 10.0%. JPEG and the method proposed in [12] are also evaluated for comparison.

The graph of the relation between the amount of bits and PSNR is shown in Fig. 6. The coding efficiency is improved compared to the schemes of JPEG and previous method for shown bit rates. The proposed coding scheme has a comparable efficiency to JPEG or other coding methods. The result of PSNR for all images is shown in Table 1.

Fig. 7 shows the image of the output of decoding compared with previous methods for a image "Airplane". Our method can reproduce

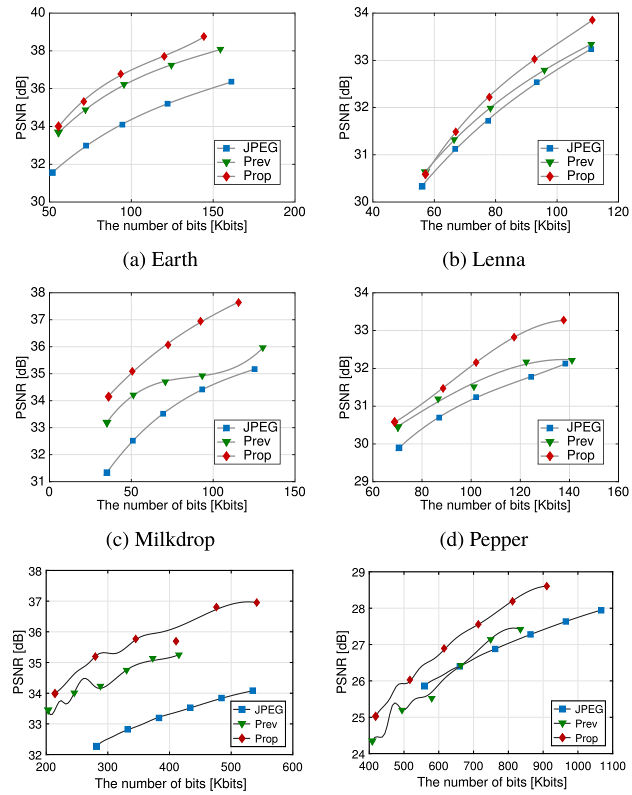
Table 1: Comparison of decorded images in PSNR [dB]

Image	Ratio [%]	JPEG	Previous	Proposed
256×256				
Airplane	4.5	29.56	30.63	**30.86**
	7.5	32.34	33.26	**33.32**
	10.0	33.89	34.52	**34.85**
Earth	4.5	31.75	33.65	**33.97**
	7.5	33.79	35.90	**36.35**
	10.0	35.21	37.24	**37.70**
Lenna	4.5	30.41	**30.65**	30.59
	7.5	32.53	32.60	**32.90**
	10.0	33.70	33.93	**34.35**
Milkdrop	2.5	31.00	32.78	**33.86**
	4.5	33.04	34.33	**35.56**
	8.0	34.42	34.93	**36.97**
Pepper	6.0	30.20	30.22	**31.04**
	8.0	31.24	31.52	**32.05**
	10.0	31.77	32.16	**32.70**
Sailboat	4.5	29.76	30.57	**30.99**
	7.0	31.59	32.02	**32.94**
	10.0	33.27	32.97	**34.34**
512×512				
Image	Ratio [%]	JPEG	Previous	Proposed
Airplane	4.0	31.89	34.20	**34.71**
	5.0	32.65	34.34	**35.32**
	6.0	33.17	35.32	**35.94**
Lenna	4.0	32.73	32.46	**33.20**
	5.0	33.31	32.89	**33.80**
	6.0	33.79	33.22	**34.21**
Mandril	7.0	25.10	24.56	**25.27**
	10.0	26.26	26.11	**27.00**
	13.0	27.10	27.34	**28.21**
Milkdrop	3.0	32.39	31.95	**35.13**
	4.0	33.19	32.00	**36.37**
	5.0	33.62	32.22	**36.96**
Pepper	4.0	29.74	29.83	**30.84**
	5.0	30.18	30.16	**31.41**
	6.0	30.50	30.42	**31.74**

blight colors correctly compared with a previous method. This is because our method introduces an adaptive colorspace instead of YCbCr colorspace. In addition, color stains in the JPEG or false colors in previous method can be reduced in our method.

6. CONCLUSION

In this paper, we proposed a color image coding scheme for approximating the chrominance signals from the luminance signals. Our method introduced the adaptive color transform to reduce the approximation error and automated the amount of segmentation to ensure the coding efficiency for given image qualities. Optimizing the colorspace is the heaviest problem for the encoder and it takes time to process. In addition, learning the color transform matrix is ill-posed and there is still an issue to minimize the approximation error and limit bits to transfer at the same time. Despite the adaptive color transform, there is a limit to the quality of image when the chrominance layer is expressed as the linear transform of luminance layers. Therefore the expression other than linear combination will be needed for better reconstruction for color image.

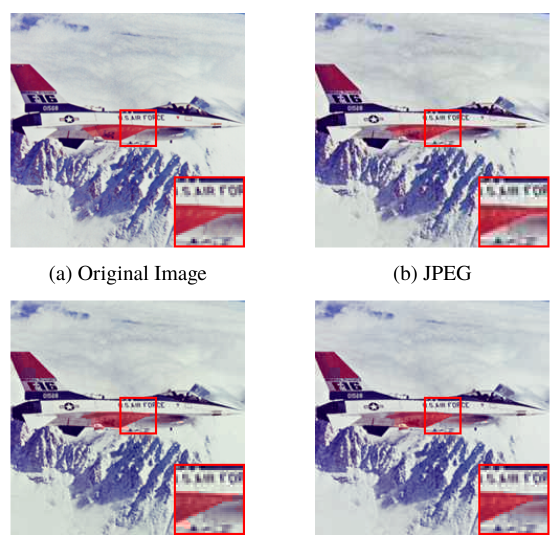


(a) Earth (b) Lenna

(c) Milkdrop (d) Pepper

(e) Airplane (512×512) (f) Mandril (512×512)

Fig. 6. The Bits-PSNR map compareing JPEG, previous, and proposed method.



(a) Original Image (b) JPEG

(c) Previous (d) Proposed

Fig. 7. Comparison of decoded outputs for image "Airplane"

7. REFERENCES

[1] "ISO/IEC 10918-1, Information technology - Digital compression and coding of continuous-tone still images Requirements and Guidelines," .

[2] Xing San, Hua Cai, and Jiang Li, "Color image coding by using inter-color correlation," in *IEEE International Conference on Image Processing*, Oct 2006, pp. 3117–3120.

[3] S. Ono, T. Miyata, and Y. Sakai, "Colorization-based coding by focusing on characteristics of colorization bases," in *Picture Coding Symposium (PCS)*, Dec 2010, pp. 230–233.

[4] T. Miyata, Y. Komiyama, Y. Sakai, and Y. Inazumi, "Novel inverse colorization for image compression," in *Picture Coding Symposium (PCS)*, May 2009, pp. 1–4.

[5] Sukho Lee, Sang wook Park, P. Oh, and Moon Gi Kang, "Colorization-based compression using optimization," *IEEE Transactions on Image Processing*, vol. 22, no. 7, pp. 2627–2636, July 2013.

[6] Li Cheng and S. V. N. Vishwanathan, "Learning to compress images and videos," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, ICML '07, pp. 161–168, ACM.

[7] Megumi Nishi, Takahiko Horiuchi, and Hiroaki Kotera, "A novel picture coding using colorization technique," *NIP and Digital Fabrication Conference*, vol. 2005, no. 2, pp. 380–383, Jan 2005.

[8] K. Uruma, K. Konishi, T. Takahashi, and T. Furukawa, "Color image coding based on the colorization algorithm using multiple resolution images," in *2015 IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2015, pp. 1290–1293.

[9] Yoshitaka Inoue, Takamichi Miyata, and Yoshinori Sakai, "Colorization based image coding by using local correlation between luminance and chrominance," *IEICE Transactions on Information and Systems*, vol. 95, no. 1, pp. 247–255, jan 2012.

[10] K. Mishiba and T. Yoshitome, "Colorization matrix construction with high compression efficiency for colorization-based coding using optimization," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5551–5555.

[11] Anat Levin, Dani Lischinski, and Yair Weiss, "Colorization using optimization," in *ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, SIGGRAPH '04, pp. 689–694, ACM.

[12] Shoji Sakamoto, Takanori Fujisawa, and Masaaki Ikehara, "Colorization-based image coding based on superpixel," in *10th International Conference on Information, Communications and Signal Processing (ICICS)*, Dec 2015.

[13] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.