A High Performance Lossless Bayer Image Compression Scheme

King-Hong Chung, Yuk-Hee Chan, Chang-Hong Fu and Yui-Lam Chan

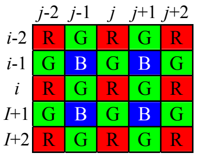
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

***Abstract*— Demosaicing and compression are generally performed sequentially in most digital cameras. Recent reports show that the compression-first scheme outperforms the conventional demosaicing-first scheme in terms of image quality and complexity. In this paper, an efficient lossless compression scheme for Bayer images is presented. It exploits a context matching technique to rank the neighboring pixels for predicting a pixel. Besides, an adaptive color difference estimation scheme is also proposed to remove the spectral redundancy. Simulation results show that the proposed algorithm can achieve a better compression performance as compared with the existing lossless CFA image coding methods.**

Index Terms—*Image Compression, Color Filter Array, Cameras*

I. Introduction

Most digital cameras use a single image sensor to capture scene images. In these cameras, Bayer color filter array (CFA) [1], as shown in Fig. 1, is usually coated over a sensor to record only one of the three chromatic components at each pixel location. In general, a CFA image is first interpolated via a demosaicing process [2-5] to form a full color image and then compressed for storage. Fig. 2a shows the workflow of this imaging chain.



	j-2	*j*-1	*j*	*j*+1	*j*+2
i-2	R	G	R	G	R
i-1	G	B	G	B	G
i	R	G	R	G	R
I+1	G	B	G	B	G
I+2	R	G	R	G	R

Fig. 1 – Bayer pattern having a red sample as its center

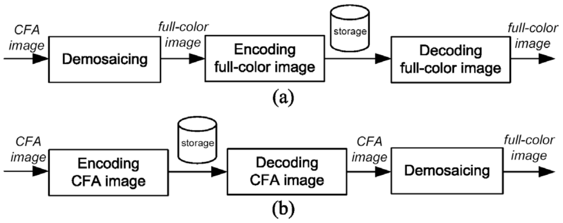


Fig. 2 – Single-sensor camera imaging chain: (a) the demosaicing-first scheme, (b) the compression-first scheme

Recently, some reports [6,7] indicated that such a demosaicing-first scheme was inefficient as the demosaicing process introduced redundancy which should eventually be removed in the compression step. As a result, an alternative approach [6,7] which carries out compression before demosaicing as shown in Fig. 2b has been proposed lately. Under this new strategy, digital camera can have a simpler design and lower power consumption is required as the computationally heavy processes like demosaicing can be carried out in an offline powerful personal computer. This motivates the demand of CFA image compression techniques.

There are two approaches for CFA image compression: lossy and lossless. Lossy compression is generally referred to as visually lossless compression as it only discards visually redundant information. This approach usually yields a higher compression ratio as compared with the lossless approach. [6-8] are some examples of this approach.

In some high-end photography applications, original CFA images are required for producing high quality full color images directly, and hence lossless compression of CFA images is necessary. Some grayscale image lossless compression methods like JPEG-LS [9] and JPEG2000 [10] can be used to encode a CFA image but only a fair performance can be attained. Recently, an advanced lossless CFA image compression algorithm (LCMI) [11] was proposed. In this algorithm, the mosaic data is de-correlated by the Mallat wavelet packet transform and the coefficients are then compressed by adaptive Rice code.

In this paper, a simple prediction-based lossless CFA compression scheme is presented. It employs context matching technique to rank the neighboring pixels for predicting the current pixel. In addition, an adaptive color difference estimation technique is also used to remove the spectral redundancy. Experimental results show that the proposed compression method can effectively and efficiently reduce the redundancy in both spatial and spectral domains. As compared with the existing lossless CFA image coding algorithms, the proposed scheme provides the best compression performance.

This paper is structured as follows. In the next section, the proposed context matching based prediction scheme is presented. In Sections III and IV, the description of an adaptive color difference estimation technique and the structure of the proposed compression scheme are, respectively, provided. Section V demonstrates some simulation results and, finally, a conclusion is given in Section VI.

II. Context Matching Based Prediction

The proposed prediction scheme handles the green plane and the non-green planes separately in a raster scan manner. It reorders the neighboring samples such that the one has higher context similarity to that of the current sample will contribute more to the current prediction.

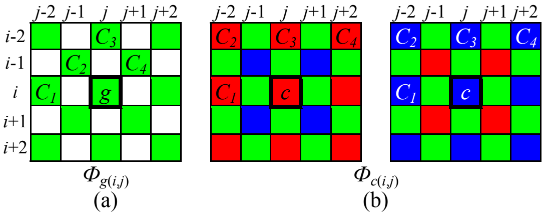


Fig. 3 – Positions of the pixels included in the candidate set of (a) a green sample and (b) a red/blue sample

1-4244-1437-7/07/$20.00 ©2007 IEEE

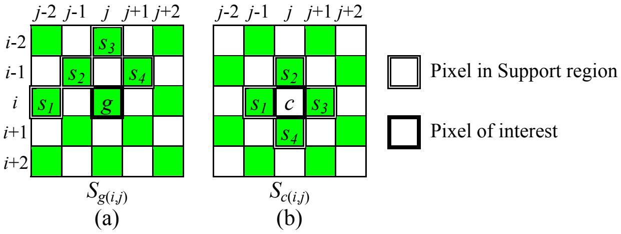


Fig. 4 – The support region of (a) a green sample and (b) a red/blue sample

Let us consider the prediction on the green plane first. Assume that we are now processing a particular green sample $g(i,j)$ as shown in Fig 3a. The four nearest neighboring green samples of $g(i,j)$ form a candidate set $\Phi_{g(i,j)}$={$g(i,j$-2), $g(i$-1,j-1), $g(i$-2,j), $g(i$-1,j+1)}. The candidates are ranked by comparing their support regions (i.e. context) with that of sample $g(i,j)$. The support region of a green sample at position (i,j), $S_{g(i,j)}$, is defined as shown in Fig. 4a. In formulation, we have $S_{g(i,j)}$={(i,j-2), (i-1,j-1), (i-2,j), (i-1,j+1)}. The matching extent of the support regions of $g(i,j)$ and $g(m,n)$ for $g(m,n) \in \Phi_{g(i,j)}$ is then measured by

$$D(S_{g(i,j)},S_{g(m,n)}) = \left|g_{(i,j-2)} - g_{(m,n-2)}\right| + \left|g_{(i-1,j-1)} - g_{(m-1,n-1)}\right| + \left|g_{(i-2,j)} - g_{(m-2,n)}\right| + \left|g_{(i-1,j+1)} - g_{(m-1,n+1)}\right| \quad (1)$$

Let $g(m_k,n_k) \in \Phi_{g(i,j)}$ for k=1,2,3,4 be the 4 ranked candidates of sample $g(i,j)$ such that $D(S_{g(i,j)},S_{g(m_u,n_u)}) \le D(S_{g(i,j)},S_{g(m_v,n_v)})$ for $1 \le u < v \le 4$. The value of sample $g(i,j)$ can then be predicted with a prediction filter as

$$\hat{g}(i,j) = round\left(\sum_{k=1}^{4} w_k g(m_k,n_k)\right) \quad (2)$$

where w_k for k=1,2,3,4 are weighting coefficients. In our study, w_k's are obtained by quantizing the training result derived by linear regression with a set of training images covering half of the test images shown in Fig. 6. They are quantized to reduce the realization effort of eqn. (2). After all, the coefficients of the prediction filter used to obtain the result presented in this paper are {w_1, w_2, w_3, w_4}={5/8, 2/8, 1/8, 0} and the predicted green sample is given by

$$\hat{g}(i,j) = round\left(\frac{4g(m_1,n_1) + 2g(m_2,n_2) + g(m_1,n_1) + g(m_3,n_3)}{8}\right). \quad (3)$$

As for the case when the sample being processed is a red or blue sample, the prediction is carried out in the color difference domain instead of the green color plane as before.

Let $d(p,q)$ be the green-red (or green-blue) color difference value of a non-green sample $c(p,q)$. Its determination will be discussed in detail in Section III, where an adaptive color difference estimation method is proposed.

For any non-green sample $c(i,j)$, its candidate set becomes $\Phi_{c(i,j)}$={$d(i,j$-2), $d(i$-2,j-2), $d(i$-2,j), $d(i$-2,j+2)} and its support region (context) is defined as $S_{c(i,j)}$={(i,j-1), (i-1,j), (i,j+1), (i+1,j)} as shown in Fig. 3b and Fig. 4b, respectively.

The prediction for a non-green sample is carried out in color difference domain. Specifically, the predicted color difference value of sample $c(i,j)$ is given by

$$\hat{d}(i,j) = round\left(\sum_{k=1}^{4} w_k d(m_k,n_k)\right) \quad (4)$$

where w_k and $d(m_k,n_k)$ are, respectively, the k^{th} predictor coefficient and the k^{th} ranked candidate in $\Phi_{c(i,j)}$ such that $D(S_{c(i,j)},S_{c(m_u,n_u)}) \le D(S_{c(i,j)},S_{c(m_v,n_v)})$ for $1 \le u < v \le 4$, where

$$D(S_{c(i,j)},S_{c(m,n)}) = \left|g_{(i,j-1)} - g_{(m,n-1)}\right| + \left|g_{(i,j+1)} - g_{(m,n+1)}\right| + \left|g_{(i-1,j)} - g_{(m-1,n)}\right| + \left|g_{(i+1,j)} - g_{(m+1,n)}\right| \quad (5)$$

Again, w_k are trained with the same set of training images used to train the predictor coefficients in eqn.(2). For the compression results reported in this paper, the predictor used for the color difference prediction is

$$\hat{d}(i,j) = round\left(\frac{4d(m_1,n_1) + 2d(m_2,n_2) + d(m_3,n_3) + d(m_4,n_4)}{8}\right). \quad (6)$$

In the proposed compression scheme, all green, red and blue pixels are encoded respectively in a raster scan manner. The four samples used for predicting sample $g(i,j)$ in eqn.(2) are $g(i,j)$'s closest processed neighboring samples of the same color. They have the highest correlation to $g(i,j)$ in different directions and hence can provide a good prediction result even in an edge region. A similar argument applies to explain why $\Phi_{c(i,j)}$ is used in eqn.(4) to handle blue/red samples.

As for the support regions, no matter the concerned sample is green or not, as shown in Fig. 4, their supports are defined based on their four closest known green samples. This is because the green channel has a double sampling rate as compared with the other channels in a CFA image and hence provides a more reliable context for matching. In the proposed scheme, as green samples are encoded first in raster sequence, all green samples are known in the decoder and hence the support of a non-green sample can be non-causal while the support of a green sample has to be causal. This non-causal support tightly and completely encloses its sample of interest. It models image features such as intensity gradient, edge orientation and textures better such that more accurate support matching can be achieved.

III. Adaptive Color Difference Estimation

When compressing the non-green color plane, color differences are exploited to remove the spectral dependency. Let $c(m,n)$ be the intensity value of the available color (either red or blue) at a non-green sampling position (m,n). The green-red (green-blue) color difference of pixel (m,n), $d(m,n)$, is obtained by

$$d(m,n) = \hat{g}(m,n)\text{-}c(m,n) \quad (7)$$

where $\hat{g}(m,n)$ represents an estimate of the missing green component at position (m,n). In the proposed estimation, $\hat{g}(m,n)$ is adaptively determined according to the horizontal gradient δH and the vertical gradient δV at (m,n) as follows.

$$\hat{g}(m,n) = round\left(\frac{\delta H \times G_V + \delta V \times G_H}{\delta H + \delta V}\right) \quad (8)$$

where $G_H = \frac{g(m,n-1) + g(m,n+1)}{2}$ and $G_V = \frac{g(m-1,n) + g(m+1,n)}{2}$

donate, respectively, the preliminary green estimates obtained by linearly interpolating the adjacent green samples horizontally and vertically. Note that, in eqn.(8), the missing green value is determined in such a way that a preliminary estimate contributes less if the gradient in the corresponding direction is larger. The weighing mechanism automatically directs the estimation process along an edge if there is.

Gradients δH and δV are determined by averaging all local green gradients in the same direction within a 5×5 window as

$$\delta H = \frac{1}{5} \sum_{\substack{(p,q)\in\{(m-1,n-2),\\(m+1,n-2),(m,n-1),\\(m-1,n),(m+1,n)\}}} \left| g(p,q) - g(p,q+2) \right|$$

and

$$\delta V = \frac{1}{5} \sum_{\substack{(p,q)\in\{(m-2,n-1),\\(m-2,n+1),(m-1,n),\\(m,n-1),(m,n+1)\}}} \left| g(p,q) - g(p+2,q) \right| \cdot \qquad (9)$$

IV. Proposed Compression Scheme

Fig. 5 shows the proposed compression scheme. In the encoding phase, a CFA image is first divided into 2 sub-images: a green sub-image which contains all green samples of a CFA image and a non-green sub-image which holds the rest samples. The green sub-image is coded first and the non-green sub-image follows based on the green sub-image as a reference.

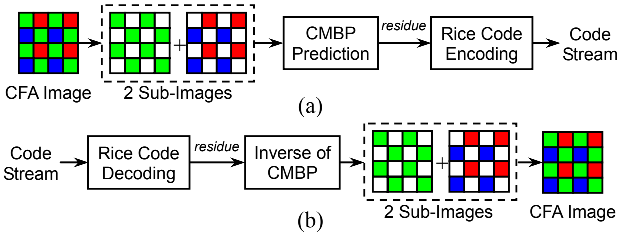


Fig. 5 – Structure of the proposed compression scheme: (a) encoder and (b) decoder

To code a sub-image, the sub-image is raster-scanned and each pixel is predicted with its 4 neighboring pixels by using context matching based prediction (CMBP), the prediction scheme proposed in Section II. Its prediction error, say $e(i,j)$, is given by

$$e(i,j) = \begin{cases} \hat{g}(i,j) - g(i,j) & \text{when } (i,j) \text{ is in green sub-image} \\ \hat{d}(i,j) - d(i,j) & \text{when } (i,j) \text{ is in non-green sub-image} \end{cases} \qquad (10)$$

where $g(i,j)$ is the real green sample value (if it exists) and $d(i,j)$ is the color difference of pixel (i,j) estimated by the method described in Section III. $\hat{g}(i,j)$ and $\hat{d}(i,j)$, respectively, represent the predicted value of $g(i,j)$ and the predicted value of $d(i,j)$. The error residue $e(i,j)$ is then mapped to a non-negative integer as follows to reshape its value distribution from a Laplacian one to an exponential one for Rice code.

$$E(i,j) = \begin{cases} 2e(i,j) & \text{if } e(i,j) \geq 0 \\ -2e(i,j) - 1 & \text{otherwise} \end{cases} \qquad (11)$$

The Rice code is employed in the proposed scheme because of its simplicity and its efficiency in handling exponentially distributed sources. In Rice code, the mapped residue $E(i,j)$ is compressed by splitting it into a quotient Q=floor($E(i,j)/\lambda$) and a remainder $R=E(i,j)\bmod\lambda$ with a positive integer $\lambda=2^k$ called Rice parameter, where $k\geq 0$. The quotient and the remainder are then saved for storage or transmission. The length of the codeword used for representing $E(i,j)$ is given by

$$L(E(i,j) \mid \lambda) = floor\left(\frac{E(i,j)}{\lambda}\right) + 1 + \log_2 \lambda \cdot \qquad (12)$$

Parameter λ is critical to the compression performance of Rice code as it determines the code length of $E(i,j)$. In the proposed scheme, parameter λ is determined by $\theta(i,j)$, a local information index of pixel (i,j) defined as

$$\theta(i,j) = E(m_1,n_1) + E(m_2,n_2) + \frac{E(m_3,n_3) + E(m_4,n_4)}{2} \qquad (13)$$

where $E(m_k,n_k)$ for k=1,2,3,4 are the corresponding mapped prediction errors of the four ranked candidates in $\Phi_{g(i,j)}$ (or $\Phi_{c(i,j)}$, depends on whether (i,j) is a green sample) acquired during CMBP. As the residue plane still contains certain amount of pixel correlation, applying the ranked local information to compute $\theta(i,j)$ can better model residue $E(i,j)$ for the estimation of parameter λ.

$\theta(i,j)$ can then be classified into a few of groups, and the corresponding parameter λ for $E(i,j)$ is determined as

$$\lambda(i,j)=2^p, \quad \text{where} \quad p = \arg_z \{ b_{z-1} \leq \theta(i,j) < b_z \} \qquad (14)$$

In (14), b_z are the boundaries of a K-class classifier $\boldsymbol{C}=\{0\equiv b_0 < b_1 < \cdots < b_K \equiv +\infty\}$ for z=1,2,⋯,K. They are determined by minimizing the total code length for the sub-image using dynamic programming, i.e.

$$\boldsymbol{C}_{opt} = \min_{C} \sum_{(i,j)\in\zeta} L(E(i,j) \mid \lambda(i,j)) \qquad (15)$$

where ζ donates the set of all pixels in a sub-image.

As it is not practical to compute b_z on the fly, they are obtained by offline training. With the same set of training images mentioned in Section II, classifiers $\boldsymbol{C}_g$={0, 1, 8, 21, 43, 99, 251, +∞} and $\boldsymbol{C}_c$={0, 1, 8, 21, 47, 118, 326, 574, +∞} were obtained and used for coding the green sub-image and the non-green sub-image respectively in our simulations.

The decoding process is just the reserve process of encoding. The green sub-image is decoded first and then the non-green sub-image is decoded with the decoded green sub-image as a reference. The original CFA image is then reconstructed by combining the two sub-images.

V. Compression Performance

Simulations were carried out to evaluate the compression performance of the proposed scheme. Twenty-four 24-bit color images of size 512×768 each as shown in Fig. 6 were Bayer sub-sampled to form a set of 8-bit testing CFA images. They were then directly coded by the proposed compression scheme. Some representative lossless compression algorithms such as JPEG-LS [9], JPEG 2000 (lossless mode) [10] and LCMI [11] were also evaluated for comparisons.

Table 1 lists the output bit-rates of the CFA images achieved by various algorithms. It clearly shows that the proposed scheme outperforms all other evaluated methods in all testing images. Especially for the images which contain many fine textures such as images 5, 7, 8, 13, 20 and 24, the bit-rates achieved by the proposed scheme are at least 0.32bpp lower than the corresponding bit-rates achieved by LCMI, which is the method offers the second best compression performance. On average, the proposed scheme yields a bit-rate as low as 4.64bpp. It is, respectively, around 1.26, 0.37 and 0.25 bpp lower than those achieved by JPEG-LS, JPEG2000 and LCCMI.

Image	JPEG -LS	JPEG 2000	LCMI	Ours	Image	JPEG-LS	JPEG 2000	LCMI	Ours
1	6.403	5.816	5.824	5.517	**13**	6.747	6.372	6.503	6.181
2	6.787	5.134	4.629	4.331	**14**	6.289	5.555	5.487	5.191
3	5.881	4.216	3.965	3.761	**15**	6.317	4.656	4.396	4.092
4	6.682	4.931	4.606	4.385	**16**	5.289	4.552	4.521	4.417
5	6.470	5.947	5.859	5.402	**17**	4.965	4.547	4.499	4.291
6	5.871	5.210	5.139	4.918	**18**	6.184	5.570	5.538	5.323
7	5.974	4.500	4.299	3.975	**19**	5.470	4.909	4.898	4.777
8	6.295	5.899	5.966	5.634	**20**	4.317	4.026	4.054	3.550
9	5.074	4.391	4.319	4.201	**21**	5.467	5.039	4.983	4.830
10	5.395	4.556	4.415	4.221	**22**	6.188	5.218	5.060	4.873
11	5.370	4.986	4.952	4.709	**23**	6.828	4.525	3.960	3.856
12	5.628	4.485	4.307	4.107	**24**	5.719	5.223	5.257	4.915
					Avg.	**5.900**	**5.011**	**4.893**	**4.644**

Table 1 – Achieved bit-rates of various lossless compression algorithms in terms of bits per pixel (bpp)

As a green pixel estimation method is proposed and used when compressing the non-green sub-image in color difference domain, a simulation was also carried out to evaluate its performance. For comparison, some other estimation methods such as bilinear interpolation [4] (BI), edge sensing interpolation [5] (ESI) and adaptive directional interpolation [2] (ADI) were also evaluated. To provide a clear demonstration, only the non-green sub-image was coded in this simulation.

Table 2 reveals the average bit rates of the outputs achieved by various estimation algorithms. It shows that the proposed adaptive estimation method yields the best compression performance among the evaluated estimation methods. On average, the proposed estimation method achieves a bit-rate of 4.51bpp which is around 0.08 bpp lower than that achieved by BI.

Image	BI	ESI	ADI	Ours	Image	BI	ESI	ADI	Ours
1	5.439	5.438	5.373	5.267	**13**	6.024	6.021	6.038	5.970
2	4.373	4.369	4.373	4.327	**14**	5.053	5.051	5.026	4.958
3	3.745	3.742	3.731	3.689	**15**	4.190	4.182	4.165	4.147
4	4.333	4.330	4.323	4.297	**16**	4.322	4.320	4.301	4.204
5	5.305	5.293	5.259	5.192	**17**	4.210	4.208	4.173	4.142
6	4.875	4.873	4.826	4.731	**18**	5.167	5.165	5.178	5.137
7	3.885	3.887	3.879	3.851	**19**	4.731	4.729	4.659	4.594
8	5.583	5.577	5.462	5.328	**20**	3.717	3.713	3.693	3.684
9	4.164	4.166	4.127	4.092	**21**	4.767	4.765	4.753	4.704
10	4.174	4.174	4.145	4.091	**22**	4.763	4.763	4.762	4.724
11	4.642	4.635	4.607	4.511	**23**	3.860	3.860	3.855	3.845
12	4.092	4.090	4.071	3.998	**24**	4.806	4.804	4.793	4.694
					Avg.	**4.592**	**4.590**	**4.565**	**4.507**

Table 2 – Achieved bit-rates (bpp) for coding non-green planes with the proposed coding scheme using various estimation methods to estimate a green sample for reference

Fig. 6 – Twenty-four digital color images (Refers as image 1 to image 24, from top-to-bottom and left-to-right)

VI. Conclusion

In this paper, a lossless compression scheme for Bayer images is proposed. This scheme separates a CFA image into a green sub-image and a non-green sub-image and then encodes them separately with prediction coding. The prediction is carried out in the intensity domain for the green sub-image while it is carried out in the color difference domain for the non-green sub-image. In both cases, a context matching technique is used to rank the neighboring pixels of a pixel for predicting the existing sample value of the pixel. The prediction residues of the two sub-images are then separately encoded with Rice code adaptively.

Experimental results show that the proposed compression scheme can efficiently and effectively decorrelate the data dependency in both spatial and color spectral domains. Consequently, it provides the best average compression ratio as compared with the latest lossless Bayer image compression schemes.

VII. Acknowledgement

This work was supported by the RGC of Hong Kong Special Administrative Region (PolyU5205/04E) and The Hong Kong Polytechnic University (PolyU Grant G-YE90).

References

[1] E. Bayer, "Color imaging array", U.S. Patent 3 971 065, July 1976.

[2] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer and R.M. Mersereau, "Demosaicking: color filter array interpolation", IEEE Signal Processing Magazine, vol.2, no. 1, pp.44-54, Sept. 2005.

[3] K. H. Chung and Y. H. Chan, Color Demosaicing Using Variance of Color Differences, IEEE Trans. on Image Processing, Vol. 15, No.10, pp.2944-2955, Oct 2006.

[4] T. Sakamoto, C. Nakanishi and T. Hase, "Software pixel interpolation for digital still camera suitable for a 32-bit MCU", IEEE Trans. Consumer Electronics, vol. 44, no. 4, pp.1342-1352, Nov. 1998.

[5] R. Lukac and K. N. Plataniotis, "Data-adaptive filters for demosaicking: A framework", IEEE Trans. Consumer Electronics, vol. 51, no. 2, pp.560-570, May 2005.

[6] X. Xie, G.L. Li, Z. H. Wang, C. Zhang, D.M. Li, X.W. Li, "A novel method of lossy image compression for digital image sensors with Bayer color filter arrays", Int. Symp. Circuits and Systems Proc. 2005, vol. 5, pp.4995-4998, May 2005.

[7] C. C. Koh, J. Mukherjee and S. K. Mitra, "New efficient methods of image compression in digital cameras with color filter array", IEEE Trans. Consumer Electronics, vol. 49, no. 4, pp.1448-1456, Nov. 2003.

[8] A. Bruna, F. Vella, A. Buemi and S. Curti, "Predictive differential modulation for CFA compression", Signal Processing Symp. Proc. 2004, pp.101-104, Jun. 2004.

[9] ISO/IEC JTC1/SC29/WG1, "Information technology – lossless and near-lossless compression of continuous-tone still image", ISO FDIS 14495-1 (JPEG-LS), 1998.

[10] ISO/IEC JTC1/SC29/WG1 N1890, "JPEG2000 Part I final draft international standard", 2000.

[11] N. Zhang and X. Wu, "Lossless compression of color mosaic images", IEEE Trans. Image Processing, vol.15, no. 6, pp.1379-1388, Jun. 2005.