Producing Color-Indexed Images with Scalable Color and Spatial Resolutions

Yik-Hing Fung and Yuk-Hee Chan

Centre for Signal Processing
Department of Electronic and Information Engineering
The Hong Kong Polytechnic University, Hong Kong

***Abstract* - Making the media content to be delivered over the Internet to clients of various capabilities scalable is crucial to improve the efficiency of various involved parties. GIF is one of the primary Web file formats used in Web applications nowadays and hence it is desirable to make them scalable as well. However, as GIF format is purposely designed for supporting clients of limited decoding capability, the scalability should be introduced with only a small increase in decoder complexity and bitstream overhead. This paper presents an algorithm that is able to produce color-indexed images with scalable color and spatial resolutions. Downscaled versions of an color-indexed image can be obtained by just bit-cropping and/or down-sampling the index plane of the image.**

1. INTRODUCTION

GIF[1] is a color-indexed image file format in which color quantization[2] is exploited. A color quantized image (a.k.a. color-indexed image) consists of a color index plane and a palette C that confines the colors of the image. To minimize the decoder complexity and the bitstream overhead, we purposely select the following approaches to extract degraded versions of the image for supporting spatial scalability and quality scalability. In particular, it reduces the spatial resolution by down-sampling the index plane and reduces the color resolution by chopping the n least significant bits of an index off to form a new index to select a color from a palette whose size is $1/(2^n)$ of that of C. Fig. 1 shows an example in which the spatial resolution of the image is reduced by 2 along each direction and the palette size is reduced to $2^5=32$.

The issue is then whether the visual quality of the downgraded versions obtained by these approaches is up to a certain standard to make the scalability meaningful especially when digital halftoning technique [3,4] is used to improve the visual quality of the full-scale color-indexed image.

To our best knowledge, a joint implementation of the aforementioned low-complexity realization of spatial and quality scalabilities has never been explicitly reported. However, there are still some promising reports in which only spatial scalability or quality scalability is concerned.

Goldschneider et al. addressed the quality scalability issue in [5]. They organize the color palette based on the minimum cost perfect matching (MCPM) technique [6] such that a low bit-depth color-indexed image can be embedded into a higher bit-depth color-indexed image. However, since a conventional error diffusion algorithm is exploited to diffuse the color error in a predefined scanning order, a pixel's color error cannot be handled until the pixel is encountered in the course, and only a very limited number of fixed neighboring pixels can help to hide the color error. This lack of flexibility forces the error to accumulate along the scanning path. Eventually it makes the compensation fail and degrades the quality of the output.

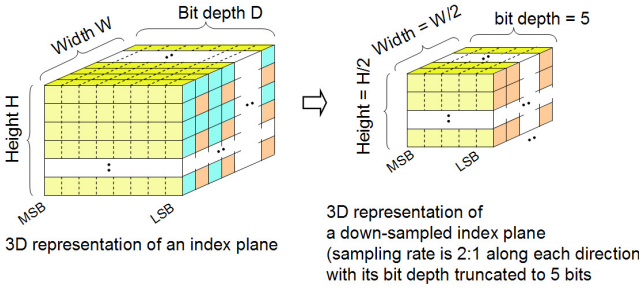

Fig. 1. An example of our targeted implementation of spatial and quality scalability for color-indexed images.

The spatial scalability of a color-indexed image is extensively studied in [7] by Fung et al. A color quantization framework is proposed in [7] to generate a spatially scalable color-indexed image based on the multiscale error diffusion framework proposed in [8] such that images of lower resolutions can be directly obtained by down-sampling. They also extend Orchard et al.'s[3] and Riemersma's[9] color quantization algorithms to produce spatially scalable color-index images in their study. However, the produced images are not quality scalable.

In this paper, we propose an algorithm that is able to produce color-indexed images with scalable color and spatial resolutions. The organization of this paper is as follows. Section 2 presents an algorithm that can turn any given palette into a binary tree-structured palette to support the production of a scalable color-indexed image. Section 3 presents the production algorithm. Section 4 provides some simulation results for evaluation. A conclusion is provided in Section 5. Fig. 2 shows the

notations used in the paper for reference. Color palettes and images are all defined in the YIQ color space.

2. Palette Generation

To support color resolution scalability, a binary tree-structured color palette having the properties shown in Fig. 3 is required. (Here we assume that the full-scale palette C is of size N_c =256.) However, color palette C may not be generated with a binary tree-structured vector quantization codebook generation algorithm [10] and hence downscaled palettes C_n's may not be well-defined. In that case, downscaled palettes C_n's have to be derived based on palette C and the original full-color image X. The details of its implementation are as follows.

Starting with palette C, we iteratively construct palette C_{k-1} based on palette C_k by merging palette colors until palette C_0 is obtained. The details are as follows.

Step 1: Merge any 2 palette colors in C_k to form a new color

$$\hat{C}_k(i,j) = \left(w_i \vec{C}_k(i) + w_j \vec{C}_k(j)\right) \Big/ \left(w_i + w_j\right) \quad \text{for } 0 \le i,j < 2^k\text{-1 and } i \ne j \qquad (1)$$

where $$w_l = \sum_{\vec{c} \in \Omega_l^k} \frac{1}{\left\| \vec{c} - \vec{C}_k(l) \right\|^2 + 0.01} \qquad (2)$$

In eqn. (2), $\vec{c}$ is the color of a pixel in X. The cost associated with this merging is given as

$$J_k(i,j) = p_i \| \vec{C}_k(i) - \hat{C}_k(i,j) \|^2 + p_j \| \vec{C}_k(j) - \hat{C}_k(i,j) \|^2 \qquad (3)$$

where p_l is the total number of pixels in X whose colors are in Ω_l^k.

Step 2: Sort $\hat{C}_k(i,j)$ for $0 \le i,j < 2^k$-1 and $i \ne j$ according to $J_k(i,j)$ in ascending order. Let the sorted sequence of $\hat{C}_k(i,j)$ be S.

Step 3: Let $\hat{C}_k(m,n)$ be the first element in S. Pick $\hat{C}_k(m,n)$ as a codeword of palette C_{k-1} and Remove all $\hat{C}_k(m,j)$, $\hat{C}_k(j,m)$, $\hat{C}_k(n,j)$ and $\hat{C}_k(j,n)$ for all j in sequence S.

Step 4. If sequence S is not empty, go back to step 3.

After all palettes are defined, the indices of their codewords are then reassigned such that codewords are properly indexed in a way as shown in Fig. 2 to form a balanced binary tree-structured color palette.

3. Production of Scalable Images

Suppose one wants to generate a scalable color-indexed image for a full-color image X by embedding various downscaled versions of the full-scale color-indexed image in it such that these scaled versions can be obtained directly by subsampling or using chopped index values to access a smaller palette as described in Fig. 1. Without loss of generality, we assume that X is of size

- X is the original 24-bit full-color image.
- X_d subsampled version of X, where d is the downsampling rate. We have $X_1 = X$.
- $I_{d(n)}$ is the index plane of our color-quantization output of X_d, where n is the bit depth of an index value and d is the downsampling rate.
- $I_{d(n)}(x,y)$ is the $(x,y)^{\text{th}}$ element of $I_{d(n)}$. It is a scalar index value.
- $X_d(x,y)$ is the $(x,y)^{\text{th}}$ element of X_d. It is a color vector.
- C is the full-scale palette (codebook). Its size is N_c.
- C_n is a downscaled palette of C. It is a codebook having 2^n codewords (i.e. size $= 2^n$). We have $C_{\log_2 N_c} = C$.
- $\vec{C}_n(k)$ is the ^{kth} codeword of palette C_n.
- Ω_k^n is the Voronoi region associated with codeword $\vec{C}_n(k)$.

Fig. 2. Notations used in the paper.

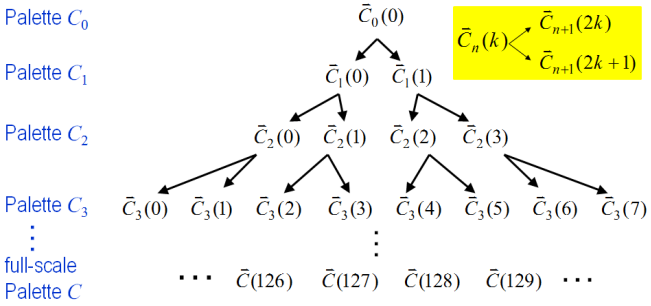

Fig.3 Codewords form a balanced binary tree-structured codebook

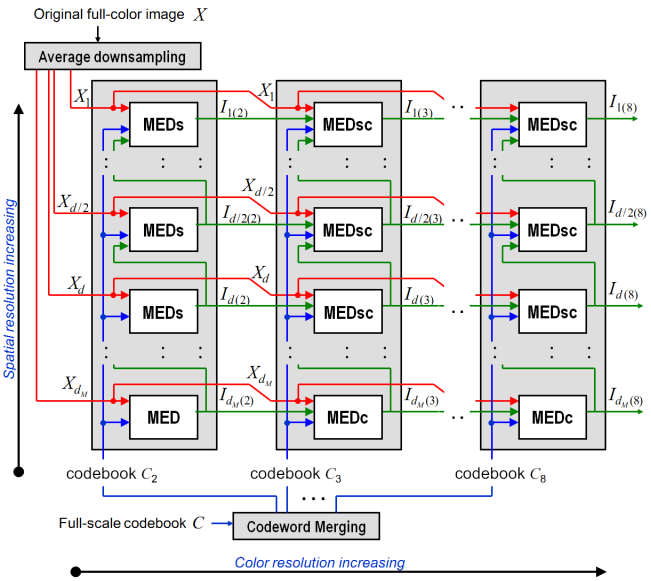

Fig. 4. How a color and spatial scalable color-indexed image is constructed by embedding its downscaled versions to the full-scale version. In this realization, the full-scale version exploits a codebook of size 256. The downscaled versions exploit codebooks of size 4, 8, 16 … and 128. The downsampling rates of the downscaled version are 2, 4, 8… and d_M.

$N \times N$, where N is an integer power of 2. The supported subsampling rate is d=2,4,…d_M, where d_M is the maximum supported subsampling rate which is a positive integer power of 2, and the supported palette size is 2^n, where n=2,3…8. In other words, the full-scale palette contains 256 colors. We note that the assumed setting is for illustration purpose only. The supported subsampling rates and palette sizes can adapt to the requirement of specific applications.

Fig. 4 shows how such a scalable color-indexed image is constructed with the given full-color image X in our proposed approach. The full-scale palette C can be a palette optimized for X or any given palette which is defined according to the physical constraints of the clients. If necessary, the downscaled palettes can be derived based on C and X as specified in Section 2. The proposed approach produces the color-indexed image of the lowest color and spatial resolution first. Versions of different spatial and color resolutions are then iteratively constructed based on versions of lower spatial and color resolutions. Four different schemes are used at different stages in the production. In all schemes, pixels are selected and processed one by one to produce their color index values. These color index values are recorded in an index plane B one by one until the whole index plane is obtained. The details of the four schemes are given as follows.

Scheme MED

Input: (i) Downsampled original having the lowest spatial resolution, X_{d_M}, and (ii) palette C_2

1. Initialize error plane $E = X_{d_M}$.
2. Select a not-yet-processed pixel in B based on the maximum error intensity guidance (MEIG) principle. Let the selected pixel be (x,y).
3. Color quantize $E(x,y)$ with palette C_2 to produce $B(x,y)$ based on the minimum Euclidean distance criterion.
4. Diffuse the color quantization error of $E(x,y)$ to update E.
5. If not all pixels are processed, goto step 2.
6. Output the index plane of B as $X_{d_M(2)}$.

Scheme MEDs

Input: (i) Downsampled original, $X_{d/2}$, (ii) index plane $I_{d(2)}$ and (iii) palette C_2

1. Initialize error plane $E = X_{d/2}$
2. For each pixel location (i,j) in $I_{d(2)}$
   a. Color quantize $E(2i,2j)$ to make $B(2i,2j)$ be the palette color of index value $I_{d(2)}(i,j)$ in palette C_2
   b. Diffuse the color quantization error of $E(2i,2j)$ to update E.
3. Select a not-yet-processed pixel in B based on the MEIG principle. Let the selected pixel be (x,y).
4. Color quantize $E(x,y)$ with palette C_2 to produce $B(x,y)$ based on the minimum Euclidean distance criterion.
5. Diffuse the color quantization error of $E(x,y)$ to update E.
6. If not all pixels are processed, goto step 3.
7. Output the index plane of B as $I_{d/2(2)}$.

Scheme MEDc

Input: (i) Downsampled original having the lowest spatial resolution, X_{d_M}, (ii) index plane $I_{d_M(n-1)}$ and (iii) palette C_n

1. Initialize error plane $E = X_{d_M}$.
2. Select a not-yet-processed pixel in B based on the MEIG principle. Let the selected pixel be (x,y) and $k = I_{d_M(n-1)}(x,y)$.
3. Color quantize $E(x,y)$ to be either $\bar{C}_n(2k)$ or $\bar{C}_n(2k+1)$ based on the minimum Euclidean distance criterion to produce $B(x,y)$.
4. Diffuse the color quantization error of $E(x,y)$ to update E.
5. If not all pixels are processed, goto step 2.
6. Output the index plane of B as $I_{d_M(n)}$.

Scheme MEDsc

Input: (i) Downsampled original, $X_{d/2}$, (ii) index planes $I_{d/2(n-1)}$ and $I_{d(n)}$, and (iii) palette C_n

1. Initialize error plane $E = X_{d/2}$
2. For each pixel location (i,j) in $I_{d(n)}$
   a. Color quantize $E(2i,2j)$ to make $B(2i,2j)$ be the palette color of index value $I_{d(n)}(i,j)$ in palette C_n.
   b. Diffuse the color quantization error of $E(2i,2j)$ to update E.
3. Select a not-yet-processed pixel in B based on the MEIG principle. Let the selected pixel be (x,y) and $k = I_{d/2(n-1)}(x,y)$.
4. Color quantize $E(x,y)$ to be either $\bar{C}_n(2k)$ or $\bar{C}_n(2k+1)$ based on the minimum Euclidean distance criterion to produce $B(x,y)$.
5. Diffuse the color quantization error of $E(x,y)$ to update E.
6. If not all pixels are processed, goto step 3.
7. Output the index plane of B as $I_{d/2(n)}$.

The algorithms for realizing the four schemes are all iterative algorithms. In each iteration, a not-yet-processed pixel in B is located via the maximum error intensity guidance (MEIG) for being color quantized. To realize MEIG, a balanced quadtree is constructed based on a masked version of the current E, say $\hat{E}$, whose $(i,j)^{\text{th}}$ element is defined as $\hat{E}(i,j) = E(i,j)M(i,j)$, where $M(i,j) = 0$ if $B(i,j)$ has been color-quantized or else it is 1. In particular, $\hat{E}$ is quadtree decomposed, and each node

corresponds to a region in $\hat{E}$. The $(m,n)^{\text{th}}$ node at level l is assigned a value of

$$\hat{E}^{l}_{(m,n)} = \left| \sum_{0\le j<N/2^{l}} \sum_{0\le j<N/2^{l}} \hat{E}\left(\frac{mN}{2^{l}} + i, \frac{nN}{2^{l}} + j \right) \right|_1$$

for $m,n = 0,1,\ldots,2^{l}-1$ (5)

where $|\bullet|_1$ denotes the L1-norm of a vector. The quadtree is searched from the root node down to a leaf to locate a pixel. During the search, it moves from the current node to its child node having the largest node value. If there are more than one qualified nodes, one of them is randomly selected.

After color-quantizing pixel (x,y), the quantization error $\vec{\varepsilon} = B(x,y) - E(x,y)$ is diffused to $E(x,y)$'s neighborhood to update image E with a non-causal diffusion filter W defined as

$$W = \begin{bmatrix} W_{(-1,-1)} & W_{(-1,0)} & W_{(-1,1)} \\ W_{(0,-1)} & W_{(0,0)} & W_{(0,1)} \\ W_{(1,-1)} & W_{(1,0)} & W_{(1,1)} \end{bmatrix} = \frac{1}{12} \begin{bmatrix} 1 & 2 & 1 \\ 2 & -12 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (6)$$

In formulation, we have

$$E(i,j) = E(i,j) - W_{(x-i,y-j)}\vec{\varepsilon} \quad \text{for } i=m\pm1 \text{ and } j=n\pm1 \quad (7)$$

When handling the boundary and the corner pixels, W is modified accordingly such that the error is not diffused out of the image boundary to avoid energy leakage.

4. Simulations

Simulations were carried out to evaluate the performance of the proposed algorithms on a number of *de facto* standard 24-bit full-color testing images including *Airplane*, *Baboon*, *Fruits*, *Peppers*, *Pool*, *Plane*, *Parrots*, *Caps*, *Barbara* and *Windows*. Each of them is of size 256×256. In the simulations, the full-scale color palette C was generated with variance-based algorithm [11] based on a particular testing image X and its size is $N_c = 128$. Downscaled palettes C_6, C_5 and C_4 were then produced with the algorithm presented in Section 2 to form a binary tree-structured codebook with $C = C_7$. The color-indexed image with the best color and spatial resolution, say Y, was constructed under an assumption that only the down-scaled versions associated with index planes $I_{d(n)}$ for $d\in\{1,2,4\}$ (i.e. d_M=4) and $n\in\{4,5,6,7\}$ would be required. Accordingly, $I_{1(7)}$ is the index plane of the full-scale color-indexed image Y in the simulation. Other $I_{d(n)}$ are obtained by bit-chopping or/and downsampling $I_{1(7)}$.

Tables 1 and 2 show the quality of the downscaled color-indexed images extracted from the full-scale color-indexed images produced with different algorithms ([5] and ours) in terms of mean square error (MSE) and the average S-CIELAB difference (Δ**E**)[12]. Each table entry shows the average quality performance for all testing images under a specific setting of d and n. We note that, through the same C is used in both evaluated algorithms in the simulation, their downscaled palettes C_6, C_5 and C_4 are different as the algorithm in [5] uses a different approach to derive the downscaled palettes.

Figures 5, 6 and 7 show portions of the downscaled color-indexed images for visual comparison. One can see that ours results are visually much better in versions of different scales. Unlike Goldschneider et al.'s algorithm [5], the proposed algorithm diffuses the color error with multiscale error diffusion instead of standard error diffusion and hence it can dilute the visual color distortion more effectively. It explains why their Y can be different as shown in Fig. 7(a) even though the same $C_7 (= C)$ is used. Besides, our algorithm takes the spatial constraint among different subsampled versions of the original image into account when diffusing the color quantization error and hence the color correlation among the pixels in a subsampled color-indexed image can be retained. Goldschneider et al.'s algorithm does not take this constraint into account, so the pixel correlation breaks down after subsampling.

Image size (=256/*d*)	[5] Palette size (=2^n)				Ours Palette size (=2^n)			
	16	32	64	128	16	32	64	128
64	0.0110	0.0114	0.0113	0.0113	0.0025	0.0020	0.0017	0.0015
128	0.0069	0.0069	0.0066	0.0066	0.0035	0.0029	0.0025	0.0024
256	0.0048	0.0046	0.0042	0.0041	0.0036	0.0029	0.0025	0.0023

Table 1. Average MSE performance of the color-indexed images of different color and spatial resolutions generated with different algorithms.

Image size (=256/*d*)	[5] Palette size (=2^n)				Ours Palette size (=2^n)			
	16	32	64	128	16	32	64	128
64	32.212	31.866	30.879	30.240	22.329	19.247	16.642	14.916
128	29.775	28.577	27.011	25.898	23.094	19.943	17.349	15.696
256	28.625	26.824	24.786	23.435	23.218	19.897	17.278	15.606

Table 2. Average S-CIELAB color difference (Δ**E**) performance of the color-indexed images of different color and spatial resolutions generated with different algorithms.

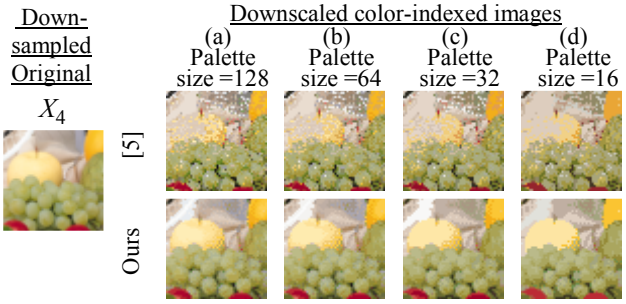

Fig. 5. Cropped regions of downscaled color-indexed images obtained with different algorithms. Their associated index planes and palettes are (a) ($I_{4(7)}$, C_7), (b) ($I_{4(6)}$, C_6), (c) ($I_{4(5)}$, C_5) and (d) ($I_{4(4)}$, C_4). Size of $I_{4(n)}$ before chopping is 64×64.

5. Conclusions

In this paper, an algorithm is proposed to generate a high-quality color-indexed image and simultaneously embed a set of color-indexed images with lower spatial and color resolutions into it. With this algorithm, images of desired color and spatial resolutions can be obtained by simple bit-cropping and/or down-sampling of its output. It

supports transmission over the Internet that contains display devices with different color resolutions, systems with different caching resources and networks with varying bandwidths and QoS capability.

The success of the algorithm highly relies on the multiscale error diffusion technique[8]. There have been new advances in the study of multiscale error diffusion recently[13-18]. A future direction of the current study would be to explore if the algorithm proposed in this paper can be further improved based on the new findings reported in [13-18].

ACKNOWLEDGEMENT

This work was supported by a grant from the Research Grants Council of the Hong Kong SAR (PolyU 5120/13E) and a grant from The Hong Kong Polytechnic University (4-ZZCZ).

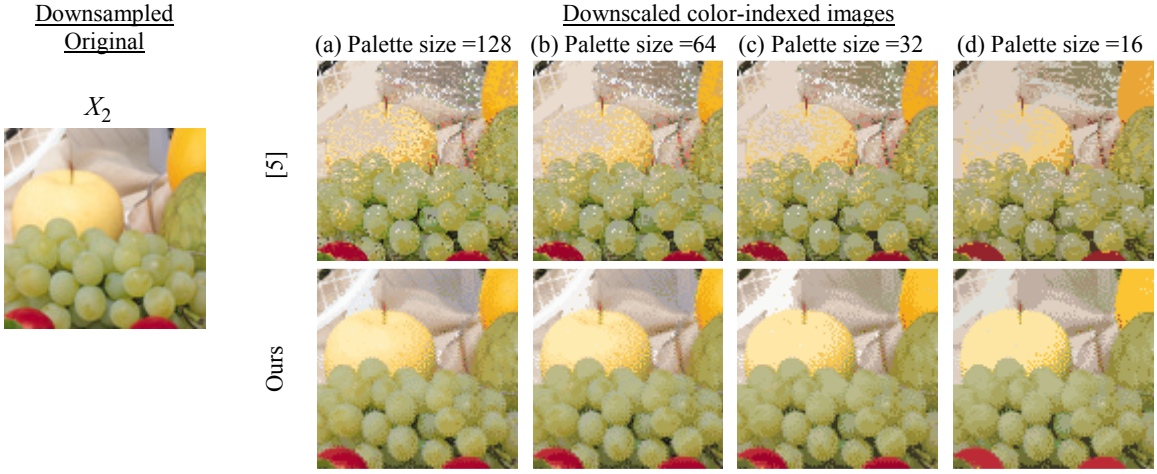

Fig. 6. Cropped regions of downscaled color-indexed images obtained with different algorithms. Their associated index planes and palettes are (a) ($I_{2(7)}$, C_7), (b) ($I_{2(6)}$, C_6), (c) ($I_{2(5)}$, C_5) and (d) ($I_{2(4)}$, C_4). Size of $I_{2(n)}$ before chopping is 128×128.

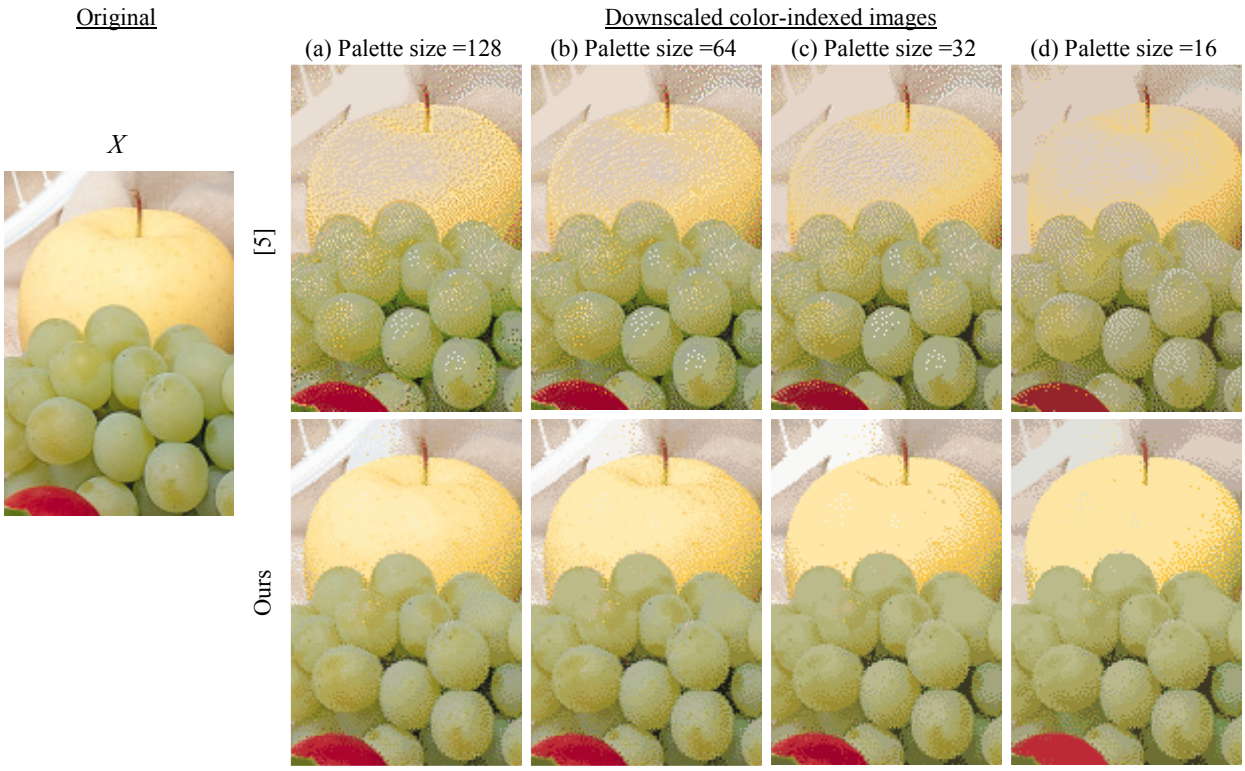

Fig. 7. Cropped regions of downscaled color-indexed images obtained with different algorithms. Their associated index planes and palettes are (a) ($I_{l(7)}$, C_7), (b) ($I_{l(6)}$, C_6), (c) ($I_{l(5)}$, C_5) and (d) ($I_{l(4)}$, C_4). Size of $I_{l(n)}$ before chopping is 256×256.

REFERENCES

[1] P. J. Lynch and S. Horton, *Web Style Guide: Basic Design Principles for Creating Web Sites*, 2nd ed. New Haven, CT: Yale Univ. Press, Mar. 2002.

[2] P. Heckbert, "Color image quantization for frame buffer displays," *Comput. Graph*., vol.16, no.4, pp.297–307, 1982.

[3] M.T. Orchard and C.A. Bouman, "Color quantization of images," *IEEE Trans. Signal Process*., vol.39, no.12, pp. 2677–2690, 1991.

[4] L. Akarun, Y. Yardimci, and A.E. Çetin, "Adaptive methods for dithering of color images," *IEEE Trans. Image Process*., vol.6, no.7, pp. 950–955, 1997.

[5] J. R. Goldschneider, E. A. Riskin and P. W. Wong, "Embedded multilevel error diffusion," *IEEE Trans. Image Process*., vol.6, no.7, pp. 956–964, 1997.

[6] D. Applegate and B. Cook, "Solving large-scale matching problems," Tech. Rep., Ctr. Discrete Math. Theoret. Comput. Scu., 1991.

[7] Y.H. Fung and Y.H. Chan, "A technique for producing scalable color-quantized images with error diffusion," *IEEE Trans. Image Process*., vol.15, no.10, pp.3218-3224, 2006.

[8] Y.H. Chan and S.M.Cheung, "Feature-preserving multiscale error diffusion for digital halftoning," *Journal of Electronic Imaging*, vol.13, no.3, pp.639-645, 2004

[9] T. Riemersma, "A balanced dither algorithm"; *C/C++ Users Journal*; vol.16, issue 12 (Dec 1998).

[10] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Norwell, MA: Kluwer, 1995.

[11] S.J. Wan, P. Prusinkiewicz, S.K.M. Wong, "Variance-based color image quantization for frame buffer display," *Color. Res. Appl.* 15 (1990) 52–58.

[12] X. Zhang and B. Wandell, "A spatial extension of cielab for digital color image reproduction," in *Proc. Soc. Information Display Dig*., San Diego, CA, 1996, pp.731–734.

[13] Y. H. Fung and Y. H. Chan, "Blue noise digital color halftoning with multiscale error diffusion," *Journal of Electronic Imaging,* 23(6), 063013, 2014, doi: 10.1117/1.JEI. 23.6.063013

[14] Y. H. Fung and Y. H. Chan, " Tone-dependent noise model for high-quality halftones," *Journal of Electronic Imaging*, 22(2), 023004, 2013. doi:10.1117/1.JEI.22.2.023004

[15] Y.H. Fung and Y.H. Chan,"Optimizing the error diffusion filter for blue noise halftoning with multiscale error diffusion," *IEEE Trans. Image Process*., vol. 22, no.1, Jan 2013, pp.413-417

[16] L.Y. Wong and Y. H. Chan, "A feature preserving multilevel halftoning algorithm," *Journal of Electronic Imaging*, 21(4), 043016, 2012. doi:10.1117/1.JEI.21.4. 043016

[17] Y.H. Fung and Y.H. Chan, "Multilevel halftoning using multiscale error diffusion," *Journal of Electronic Imaging*, vol.19, 030501, 2010. doi:10.1117/1.3479749

[18] Y.H. Fung and Y.H. Chan, "Green Noise Digital Halftoning with Multiscale Error Diffusion," *IEEE Trans. Image Process*., vol.19, no.7, Jul 2010, pp.1808-1823.