

- [13] M. Heiler and C. Schnörr, "Natural image statistics for natural image segmentation," *Int. J. Comput. Vis.*, vol. 63, no. 1, pp. 5–19, 2005.
- [14] T. Kadir and M. Brady, "Unsupervised non-parametric region segmentation using level sets," in *Proc. 9th IEEE Int. Conf. Computer Vision*, 2003, vol. 2, pp. 1267–1274.
- [15] J. Keuchel, M. Heiler, and C. Schnörr, "Hierarchical image segmentation based on semidefinite programming," in *Pattern Recognition*, C. Rasmussen, H. Bülthoff, M. Giese, and B. Schölkopf, Eds. New York: Springer, Aug. 2004, vol. 3175, pp. 120–128.
- [16] S. Kichenassamy, A. Kumar, P. Olver, A. Tannenbaum, and A. Yezzi, "Gradient flows and geometric active contour models," in *Proc. 5th Int. Conf. Computer Vision*, Jun. 1995, pp. 810–815.
- [17] Y. G. Leclerc, "Constructing simple stable descriptions for image partitioning," *Int. J. Comput. Vis.*, vol. 3, pp. 73–102, 1989.
- [18] R. Malladi, J. A. Sethian, and B. C. Vemuri, "Shape modeling with front propagation: A level set approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 2, pp. 158–175, Feb. 1995.
- [19] A. Mansouri, A. Mitiche, and C. Vázquez, "Image partitioning by level set multiregion competition," in *Proc. Int. Conf. Image Processing*, 2004, vol. 4, pp. 2721–2724.
- [20] B. Merriman, J. Bence, and S. Osher, "Motion of multiple junctions: A level set approach," *J. Comput. Phys.*, vol. 112, pp. 334–363, 1994.
- [21] D. Mumford and J. Shah, "Boundary detection by minimizing functionals. I," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Jun. 1985, pp. 22–26.
- [22] S. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, *Applied Mathematical Sciences*. New York: Springer, 2002, vol. 153.
- [23] S. Osher and N. Paragios, Eds., *Geometric Level Set Methods in Imaging, Vision and Graphics*. New York: Springer, 2003.
- [24] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, 1988.
- [25] N. Paragios and R. Deriche, "Unifying boundary and region-based information for geodesic active tracking," in *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999, vol. 2, pp. 300–305.
- [26] N. Paragios and R. Deriche, "Coupled geodesic active regions for image segmentation: A level set approach," in *Proc. 6th Eur. Conf. Computer Vision*, Dublin, Ireland, 2000, vol. 2, pp. 224–240.
- [27] N. Paragios and R. Deriche, "Geodesic active regions: A new paradigm to deal with frame partition problems in computer vision," *J. Vis. Commun. Image Represent.*, vol. 13, no. 1/2, pp. 249–268, 2002.
- [28] N. Paragios and R. Deriche, "Geodesic active regions and level set methods for supervised texture segmentation," *Int. J. Comput. Vis.*, vol. 46, no. 3, pp. 223–247, Mar. 2002.
- [29] E. Parzen, "On the estimation of a probability density function and the mode," *Ann. Math. Statist.*, vol. 33, pp. 1065–1076, 1962.
- [30] M. Rousson, T. Brox, and R. Deriche, "Active unsupervised texture segmentation on a diffusion based feature space," in *Proc. Int. Conf. Computer Vision and Pattern Recognition*, Madison, WI, Jun. 2003, pp. 699–704.
- [31] M. Rousson and R. Deriche, "A variational framework for active and adaptive segmentation of vector-valued images," in *Proc. IEEE Workshop on Motion and Video Computing*, Orlando, FL, Dec. 2002, pp. 56–62.
- [32] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "A level set model for image classification," *Int. J. Comput. Vis.*, vol. 40, no. 3, pp. 187–197, 2000.
- [33] J. A. Sethian, *Level Set Methods and Fast Marching Methods*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1999.
- [34] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
- [35] A. Tsai, A. Yezzi, and A. Willsky, "Curve evolution implementation of the Mumford-Shah functional for image segmentation, denoising, interpolation, and magnification," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1169–1186, Aug. 2001.
- [36] L. Vese and T. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *Int. J. Comput. Vis.*, vol. 50, no. 3, pp. 271–293, Dec. 2002.
- [37] H. K. Zhao, T. Chan, B. Merriman, and S. Osher, "A variational level set approach to multiphase motion," *J. Comput. Phys.*, vol. 127, pp. 179–195, 1996.
- [38] S.-C. Zhu and A. Yuille, "Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 9, pp. 884–900, Sep. 1996.

## A Technique for Producing Scalable Color-Quantized Images With Error Diffusion

Yik-Hing Fung and Yuk-Hee Chan, *Member, IEEE*

**Abstract**—To reliably and efficiently deliver media information to diverse clients over heterogeneous networks, the media involved must be scalable. In this paper, a color quantization algorithm for generating scalable color-indexed images is proposed based on a multiscale error diffusion framework. Images of lower resolutions are embedded in the outputs such that a simple down-sampling process can extract images of any desirable resolutions. Images possessing this scalable property support transmission over the Internet which contains clients with different display resolutions, systems with different caching resources and networks with varying bandwidths and QoS capabilities. Unlike most of the color halftoning algorithms available nowadays, the proposed algorithm is not dedicated for printing applications but for color-indexed displays. It works with any arbitrary palettes of different size.

**Index Terms**—Color index, color quantization, directional hysteresis, error diffusion, halftoning, multiscale processing, scalable media.

### I. INTRODUCTION

Color quantization [1] is widely used in many multimedia applications to save data storage requirement, save transmission bandwidth, and display images with a color display device that allows only a limited number of colors. When color quantization is performed, digital halftoning [2]–[4] would be helpful to improve the quality of the output by making use of the lowpass filtering property of human eyes. At the moment, the most popular halftoning method is error diffusion and several well-known error diffusion filters such as Floyd–Steinberg filter [5] and Stucki filter [6] are generally used to achieve the goal.

One of the applications of color quantization is to convert images into a color-indexed image file format called GIF. GIF is one of the two primary Web file formats used in Web applications nowadays since it inherits the benefit of fast and simple decoding [7]. When one delivers media information to diverse clients over heterogeneous networks, clients may support different display resolutions and systems may have different caching capabilities. In that case, it is desirable to make media information scalable such that it can be delivered efficiently and reliably. Since color-quantized images are widely used in multimedia applications nowadays, it is desirable to make them scalable such that

Manuscript received April 4, 2005; revised January 26, 2006. This work was supported in part by the Research Grants Council of the Hong Kong Special Administrative Region (PolyU 5205/04E) and in part by the Centre for Multimedia Signal Processing of The Hong Kong Polytechnic University (POLYU Grant A046). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Reiner Eschbach.

Y.-H. Fung and Yuk-Hee Chan are with the Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: enyhchan@polyu.edu.hk).

Color versions of Figs. 1–5 are available online at <http://ieeexplore.ieee.org>. Digital Object Identifier 10.1109/TIP.2006.877480

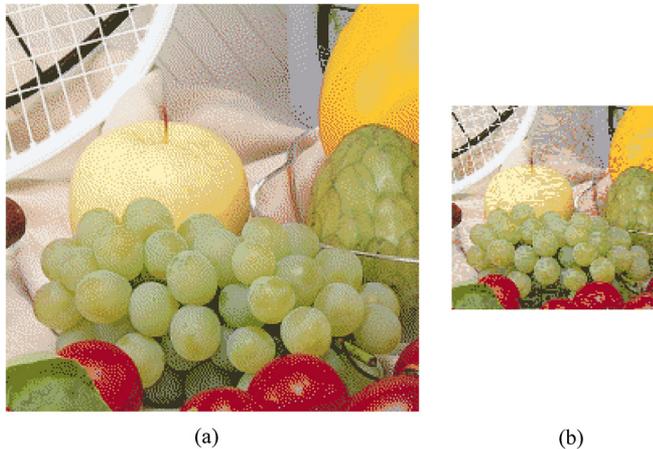


Fig. 1. Color quantization result of Orchard's algorithm [2] using a palette of size 32 and its down-sampled version. (a) Full-scaled version ( $256 \times 256$ ). (b) Down-sampled version ( $128 \times 128$ ).

their downscaled versions can be obtained directly with the images through some simple operations.

The most straightforward approach to obtain a downscaled version of a halftoned color-quantized image is down sampling. However, this approach does not work because such an image contains a lot of high-frequency energy. Fig. 1 shows the effect of directly down sampling an image produced with a conventional color quantization algorithm [2] in which error diffusion is involved. Visible artifacts can be found everywhere.

Two better approaches can be used to provide a client over heterogeneous networks a downscaled version of a halftoned color-quantized image. The first one is to generate several scaled original color images of desired sizes, color quantize each of them and store all of them in the server for future use. This is very memory-consuming since one has to store several halftoned color-quantized versions of the original color image for one single application. The second approach is to make use of postprocessing technique. In this approach, only one halftoned color-quantized image is stored. To obtain a halftoned color-quantized image of smaller size, the available halftoned color-quantized image is first restored to its original [8], [9] or low-pass filtered to remove the high-frequency noise. The processed image is then downscaled and color quantized again to produce the image of desirable resolution. This approach is computation demanding since a sequence of image processing steps has to be carried out.

The advantages of producing scalable color-quantized images are obvious. First, the generated result does not require extra memory to store a set of halftoned color-quantized images of different resolutions. Besides, only simple downsampling is required to produce the downscaled versions, and, hence, very little computational effort is required to generate this set of halftoned color-quantized images.

In general, a scalable color-quantized image can be produced by first producing an output of lower resolution and then, when producing the output of higher resolution, forcing its downscaled pixels to be the corresponding pixels in the output of lower resolution. For the sake of reference, this approach is referred to as constrained halftoning and the aforementioned pixels in the output of higher resolution are termed as constrained pixels. Similar idea was used by Wong to generate scalable binary halftones[10]. However, Wong's algorithm cannot be directly applied in producing a scalable color-quantized image.

Theoretically, any conventional binary halftoning algorithm can work with constrained halftoning to produce scalable color-quantized images after being extended to handle color-quantized images.

However, it could be more complicated than we expect. First, not all conventional binary halftoning algorithms can be directly extended to handle color-quantized images in a straightforward manner. Color quantization is actually a vector quantization instead of a bi-level uniform scalar quantization as in the case of binary halftoning. A straightforward extension of binary halftoning only works when a uniform palette is used in a color quantization process. When an arbitrary palette is used in color quantization to generate halftoned color-quantized images, modification to the algorithms is required.

Second, not all binary halftoning algorithms are suitable for constrained halftoning. Most conventional binary halftoning algorithms process pixels in a predefined scanning order. Accordingly, they do not take a constrained pixel into account until the pixel is encountered in the course. It is very likely that the value assigned to a constrained pixel is against the natural quantization result. This mismatch disturbs the harmony of a local region and degrades the quality of the output. Wong's algorithm [10] reduces this problem by using an adaptive error diffusion filter. However, pattern artifacts and directional hysteresis still exist due to the causal nature of the error diffusion filter used in this approach.

In view of the aforementioned factors, some binary error diffusion algorithms [10]–[16] would be comparatively more appropriate to work with constrained halftoning after being extended. Among them, Peli's algorithm [11] was designed to make use of the strength of both order dithering and error diffusion to produce binary halftones. Riemersma's algorithm [12] tries to reduce directional hysteresis by using a space filling curve to determine the scanning order. Aiming at removing the source of directional hysteresis, multiscale error diffusion algorithms [13]–[15] use a noncasual filter and a nonpredetermined scanning order to halftone a gray level image. All these algorithms are originally proposed for binary halftoning.

In this paper, based on the idea of constrained halftoning and multiscale error diffusion, we proposed an efficient approach to generate a halftoned color-quantized image that can be displayed at several resolutions. It is generated in such a way that, when down sampling is performed, the resultant image is also a halftoned color-quantized image and it is a high-quality rendition of the original color image at reduced resolutions. With such a scalable property, the generated halftoned color-quantized image is also suitable for progressive transmission.

## II. FRAMEWORK OF MULTISCALE ERROR DIFFUSION

This section extends the framework of binary multiscale error diffusion to take care of color quantization. In this extended framework, color quantization is performed in YIQ color space so as to reduce the correlation among different color components. Another reason for doing so is that Euclidean distance in YIQ space matches HVS response more closely as compared with that in RGB space. Processing the image in YIQ instead of RGB space allows the color quantizer to select a visually more appropriate palette color with a given input. In practice, palettes used in popular image formats such as GIF are usually defined in RGB or YIQ domain. Directly color quantizing an image in these two domains would reduce complexity and minimize the computation error introduced during format conversion. Without loss of generality, hereafter, we assume the color palette and the input image are defined in YIQ space. Though the work presented in this paper is developed in YIQ domain, it can also be developed in some other color domains such as Lab domain with the same approach.

Let  $\mathbf{X}$  be a 24-bit  $N \times N$  true-color image each pixel of which is represented as  $\vec{\mathbf{X}}_{(i,j)} = (X_{(i,j)Y}, X_{(i,j)I}, X_{(i,j)Q})$ , where  $X_{(i,j)c}$  for  $c \in \{Y, I, Q\}$  is the intensity value of the  $c$ th primary color component of the  $(i, j)$ th pixel of the image.

The proposed algorithm is an iterative algorithm. Let  $\mathbf{U}$  be an image which reports the current status of the image being processed at the beginning of a particular iteration. At each iteration, the algorithm first locates a pixel location based on the maximum energy guidance with an energy pyramid  $\mathbf{E}$  associated with  $\mathbf{U}$ . The details of the pyramid will be elaborated later. The selected pixel is then color quantized with a predefined set of colors (palette). The quantization error is diffused with a noncasual filter to neighboring pixels to update  $\mathbf{U}$ . These procedures are repeated until all pixels are color quantized. At the start of the first iteration,  $\mathbf{U}$  is initialized to be  $\mathbf{X}$ .

#### A. Constructing Energy Pyramid $\mathbf{E}$

Let  $\mathbf{M}$  be a mask of size  $N \times N$  that defines which pixels have been color quantized. Specifically, its element  $M_{(i,j)}$  is 0 if  $\tilde{\mathbf{X}}_{(i,j)}$  has been color quantized or else it is 1.

A multiscale representation of a given color image  $\mathbf{U}$  is defined as a sequence of matrices  $\{\mathbf{U}^0, \dots, \mathbf{U}^l, \dots, \mathbf{U}^L\}$ , where  $L = \log_2 N$  and  $\mathbf{U}^L = \mathbf{U}$ .  $\mathbf{U}^l$  is of size  $2^l \times 2^l$  and its  $(i,j)$ th element is a triplet  $(U_{(i,j)Y}^l, U_{(i,j)I}^l, U_{(i,j)Q}^l)$  for  $i, j = 0, 1, \dots, 2^l - 1$ . Elements of  $\mathbf{U}^l$  for  $l = 0, 1 \dots L - 2$  are defined as

$$U_{(i,j)c}^l = \sum_{m=0}^1 \sum_{n=0}^1 U_{(2i+m, 2j+n)c}^{l+1} \quad \text{for } c \in \{Y, I, Q\} \quad (1)$$

while elements of  $\mathbf{U}^{L-1}$  are defined as (2), shown at the bottom of the page, where

$$S = \sum_{m=0}^1 \sum_{n=0}^1 M_{(2i+m, 2j+n)}. \quad (3)$$

The energy pyramid  $\mathbf{E}$  associated with image  $\mathbf{U}$  is then constructed with  $\{\mathbf{E}^l | l = 0, 1, \dots, L\}$ , where  $\mathbf{E}^l$  is the energy plane of matrix  $\mathbf{U}^l$ . The  $(i,j)$ th element of  $\mathbf{E}^l$  can be defined as (4), shown at the bottom of the page, so as to match the conventional definition of energy. However, in this paper, we define it as (5), shown at the bottom of the next page, instead because the realization effort of (5) is much lower and, no matter which definition is used, the simulation results in our study is more or less the same both subjectively and objectively in terms of the S-CLELAB performance.

$$U_{(i,j)c}^{L-1} = \begin{cases} \frac{1}{S} \sum_{m=0}^1 \sum_{n=0}^1 M_{(2i+m, 2j+n)} U_{(2i+m, 2j+n)c}^L, & \text{if } S \neq 0 \quad \text{for } c \in \{Y, I, Q\} \\ 0, & \text{else} \end{cases} \quad (2)$$

$$E_{(i,j)}^l = \begin{cases} \left| (U_{(i,j)Y}^l)^2 + (U_{(i,j)I}^l)^2 + (U_{(i,j)Q}^l)^2 \right|, & \text{if } 0 \leq l < L \\ \left| M_{(i,j)} \left( (U_{(i,j)Y}^L)^2 + (U_{(i,j)I}^L)^2 + (U_{(i,j)Q}^L)^2 \right) \right|, & \text{if } l = L \end{cases} \quad (4)$$

for  $i, j = 0, 1, \dots, 2^l - 1$

$$E_{(i,j)}^l = \begin{cases} \left| U_{(i,j)Y}^l + U_{(i,j)I}^l + U_{(i,j)Q}^l \right|, & \text{if } 0 \leq l < L \\ \left| M_{(i,j)} (U_{(i,j)Y}^L + U_{(i,j)I}^L + U_{(i,j)Q}^L) \right|, & \text{if } l = L \end{cases} \quad (5)$$

for  $i, j = 0, 1, \dots, 2^l - 1$

#### B. Searching the Pixel for Color Quantization

The location of a pixel to be color quantized is determined via maximum energy guidance with energy pyramid  $\mathbf{E}$ . To a certain extent, the pixel chosen is the one with the largest energy in the neighborhood. Specifically, its location is obtained by searching the energy pyramid from the coarsest level  $\mathbf{E}^0$  to the finest level  $\mathbf{E}^L$ . Note that  $\mathbf{E}^0$  contains only one element  $E_{(0,0)}^0$ .

Assume that we are now at position  $(l, (i, j))$  which corresponds to the  $(i, j)$ th element of a particular level  $l$ . We check  $\{E_{(2i+m, 2j+n)}^{l+1} | m, n = 0, 1\}$  and proceed to the position  $(l+1, (2i+p, 2j+q))$  such that  $E_{(2i+p, 2j+q)}^{l+1}$  is the maximum in  $\{E_{(2i+m, 2j+n)}^{l+1} | m, n = 0, 1\}$ , where  $p, q \in \{0, 1\}$ . If more than one position satisfies the criterion, one of them will be randomly selected.

#### C. Color Quantization and Error Diffusion

Let  $(L, (m, n))$  be the position that we finally reach at the finest level of the pyramid  $\mathbf{E}$  in the search and  $C = \{\hat{\mathbf{v}}_i : i = 1, 2, \dots, N_c\}$  be the given color palette.  $\tilde{\mathbf{U}}_{(m,n)} = (U_{(m,n)Y}, U_{(m,n)I}, U_{(m,n)Q})$  is then color quantized. The best-matched color in the palette, say  $\hat{\mathbf{v}}_k$ , is selected based on the minimum Euclidean distance criterion in YIQ color space as follows:

$$\|\tilde{\mathbf{U}}_{(m,n)} - \hat{\mathbf{v}}_k\| \leq \|\tilde{\mathbf{U}}_{(m,n)} - \hat{\mathbf{v}}_i\| \quad \forall \hat{\mathbf{v}}_i \in C. \quad (6)$$

The quantization error  $\vec{\varepsilon} = \hat{\mathbf{v}}_k - \tilde{\mathbf{U}}_{(m,n)}$  is then diffused to  $\tilde{\mathbf{U}}_{(m,n)}$ 's neighborhood to update image  $\mathbf{U}$  with a noncasual filter. In formulation, it is given as

$$\tilde{\mathbf{U}}_{(i,j)} = \tilde{\mathbf{U}}_{(i,j)} - W_{(m-i, n-j)} \vec{\varepsilon} \quad \text{for } i = m \pm 1 \text{ and } j = n \pm 1 \quad (7)$$

where  $W$  is defined as

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



Fig. 2. Testing images.

To handle the boundary and the corner pixels,  $W$  is modified to be

$$\frac{1}{8} \begin{bmatrix} 0 & 0 & 0 \\ 2 & -8 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{and} \quad \frac{1}{5} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -5 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

respectively, to avoid energy leakage.

### III. PROPOSED MULTISCALE MULTIREOLUTION VECTOR ERROR DIFFUSION ALGORITHM

With the framework presented in the previous section, a color quantization algorithm for generating scalable color quantization images is proposed in this section.

Consider the case that one wants to produce a color quantization result of a given image  $\mathbf{I}$  in which a set of color quantization results of downsampled versions of  $\mathbf{I}$  are embedded. Let  $\mathbf{I}^r$  be one of the downsampled versions of  $\mathbf{I}$ . Without loss of generality, we assume that  $\mathbf{I}$  is of size  $N \times N$  and  $\mathbf{I}^r$  is of size  $(N/s_r) \times (N/s_r)$ , where  $s_r \in \{2^r | r = 1, 2, \dots, R; R < L = \log_2 N\}$  is a desirable scaling factor. The objective of the proposed algorithm is to produce an output  $\mathbf{Y}$  such that  $\mathbf{Y}^r$  for  $r = 1, 2, \dots, R$  can be obtained by simply down sampling  $\mathbf{Y}$ , where  $\mathbf{Y}^r$  is the color quantization result of  $\mathbf{I}^r$ .

Note that  $\mathbf{I}$  can be downsampled with any approach to obtain  $\mathbf{I}^r$ , producing different results. In this paper,  $\mathbf{I}^r$  is obtained by averaging  $\mathbf{I}$  as follows:

$$I_{(i,j)c}^r = \frac{1}{s_r \times s_r} \sum_{m=0}^{s_r-1} \sum_{n=0}^{s_r-1} I_{(s_r i+m, s_r j+n)c}$$

for  $i, j = 0, 1, \dots, (N/s_r) - 1$  and  $c \in \{Y, I, Q\}$  (8)

where  $I_{(i,j)c}^r$  and  $I_{(i,j)c}$  are, respectively, the  $c$ th color components of the  $(i, j)$ th pixels of  $\mathbf{I}^r$  and  $\mathbf{I}$ .

In the proposed algorithm, starting with  $r = R$ , we iteratively generate  $\mathbf{Y}^r$  with  $\mathbf{I}^r$  and use  $\mathbf{Y}^r$  as a constraint to produce  $\mathbf{Y}^{r-1}$  in the next iteration until  $\mathbf{Y}$  is eventually obtained.

As selected by the user,  $\mathbf{Y}^R$  is of the lowest resolution to be supported in the scalable  $\mathbf{Y}$ . There is no constraint to generate it and one can make use of the multiscale error diffusion algorithm presented in Section 2 to generate it with  $\mathbf{X} = \mathbf{I}^R$ .

To obtain  $\mathbf{Y}^r$  with  $\mathbf{I}^r$  for  $0 < r < R$ , the same multiscale error diffusion algorithm presented in Section 2 can be used by embedding a constraint in the initialization stage. Suppose one has already obtained  $\mathbf{Y}^r$  with  $\mathbf{I}^r$  and starts to produce  $\mathbf{Y}^{r-1}$  with  $\mathbf{I}^{r-1}$ . At the start of the

first iteration, after initializing  $\mathbf{U}$  to be  $\mathbf{X} = \mathbf{I}^{r-1}$ , we quantize the down-sampled elements of  $\mathbf{Y}^{r-1}$  to be

$$Y_{(2i,2j)c}^{r-1} = Y_{(i,j)c}^r \quad \text{for } i, j = 0, 1, \dots, (N/s_r) - 1 \quad (9)$$

where  $Y_{(k,l)c}^r$  is the  $c$ th color component of the  $(k, l)$ th element of  $\mathbf{Y}^r$ , and then diffuse the quantization error at positions  $(2i, 2j)$ s with (7) to update  $\mathbf{U}$ . Note assignment (9) guarantees that  $\mathbf{Y}^r$  can be obtained by simply down sampling  $\mathbf{Y}^{r-1}$ . This completes the first iteration and the following iterations are carried out as usual as it is presented in Section II until  $\mathbf{Y}^{r-1}$  is obtained.

The proposed algorithm is an iterative algorithm. Theoretically, one iteration is required to process one pixel. In each iteration,  $3 \log_2 N$  scalar comparisons are required to locate the pixel to be processed. It then takes  $N_c$  vector comparisons to select a palette color for the located pixel. After error diffusion, there are at most 9 pixels whose associated  $\tilde{\mathbf{U}}_{(m,n)}$  values are changed and, for each of these pixels,  $\log_2 N + 1$  elements of structure  $\{\mathbf{U}^0, \dots, \mathbf{U}^l, \dots, \mathbf{U}^L\}$  have to be updated. Hence, it takes two scalar multiplications and at most  $9(\log_2 N + 1)$  vector additions to update  $\{\mathbf{U}^0, \dots, \mathbf{U}^l, \dots, \mathbf{U}^L\}$ . Note this already includes the complexity of error diffusion. Finally, it takes, at most,  $18(\log_2 N + 1)$  scalar additions to update the energy pyramid  $\mathbf{E}$  with the updated  $\{\mathbf{U}^0, \dots, \mathbf{U}^l, \dots, \mathbf{U}^L\}$ . The dimension of the vectors involved in all vector operations is three. As compared with a conventional color quantization algorithm, the extra total complexity is bounded by  $O(N^2 \log_2 N)$ .

### IV. SIMULATION AND COMPARATIVE STUDY

Simulation was carried out to evaluate the performance of the algorithm on a number of *de facto* standard 24-bit full color images. Each of them is of size  $256 \times 256$ . Fig. 2 shows the testing images used in the simulation. For each testing image, a set of color palettes of different size were generated with median-cut algorithm [1] for color quantization. The proposed algorithm was applied to all testing images to obtain their corresponding halftoned color quantization results with the generated color palettes to evaluate its performance. In its realization, parameter  $R$  was selected to be 4.

For comparison, halftoned color quantization results were also produced with some other color quantization algorithms [2]–[4] and then down sampled to produce various downsampled versions. Unlike most color halftoning algorithms which are dedicated for printing applications [16], [17], these evaluated algorithms [2]–[4] are not straightforward extension of binary halftoning and are able to handle color quantization in which any arbitrary palettes can be used. Among them,

TABLE I  
AVERAGE S-CIELAB COLOR DIFFERENCE ( $\Delta E$ ) METRIC OF THE HALFTONED COLOR-QUANTIZED OUTPUTS OF VARIOUS ALGORITHMS AND THEIR DOWN-SAMPLED VERSIONS WHEN THE INVOLVED PALETTE IS OF SIZE (a) 16, (b) 32, (c) 64, AND (d) 128

Image size	Average of (Average of S-CIELAB difference $\Delta E$ )							
	Breaux [18]	Riemersma [12]	Orchard [2]	Akarun [3]	Özdemir [4]	Proposed	Riemersma+	Orchard+
(a) Palette size = 16								
Full-scaled version, $s_r = 1$	33.86	34.39	35.38	34.78	37.38	34.27	35.90	35.62
Down-sampled version, $s_r = 2$	34.66	35.53	36.70	36.42	38.63	34.09	36.11	35.61
Down-sampled version, $s_r = 3$	36.64	37.88	38.82	38.62	40.15	33.67	35.96	35.45
Down-sampled version, $s_r = 4$	40.63	41.87	42.93	42.46	43.59	31.92	33.34	33.15
(b) Palette size = 32								
Full-scaled version, $s_r = 1$	25.91	25.88	26.85	26.84	30.97	26.27	27.60	27.95
Down-sampled version, $s_r = 2$	27.26	27.46	28.85	28.84	32.16	26.24	28.02	27.97
Down-sampled version, $s_r = 3$	30.33	30.77	32.13	32.14	34.43	26.18	27.98	27.85
Down-sampled version, $s_r = 4$	35.92	36.62	37.41	37.59	38.28	24.42	25.90	25.47
(c) Palette size = 64								
Full-scaled version, $s_r = 1$	19.29	19.43	20.06	19.92	24.51	19.74	21.30	21.35
Down-sampled version, $s_r = 2$	20.93	21.53	22.46	22.39	25.85	19.86	21.83	21.48
Down-sampled version, $s_r = 3$	24.86	25.69	26.56	26.42	28.54	19.95	21.95	21.56
Down-sampled version, $s_r = 4$	31.43	32.53	33.17	32.97	33.18	18.59	19.89	19.41
(d) Palette size = 128								
Full-scaled version, $s_r = 1$	15.49	15.61	16.27	16.08	22.37	16.09	17.69	17.68
Down-sampled version, $s_r = 2$	17.74	18.11	19.06	18.90	23.55	16.38	18.28	17.97
Down-sampled version, $s_r = 3$	22.26	22.96	23.75	23.58	26.35	16.60	18.26	18.18
Down-sampled version, $s_r = 4$	29.20	30.47	30.87	30.56	30.68	15.27	15.88	15.92

Orchard's algorithm [2] forms a common framework that most of these algorithms adopt. In its realization, Floyd-Steinberg filter [5] was used in error diffusion. Both Akarun's algorithm [3] and Özdemir algorithm [4] adopt the framework presented in Orchard's algorithm. In particular, Akarun's algorithm [3] uses an adaptive error diffusion filter to prevent texture contours, color impulses and color shift. Instead of the conventional Euclidean distance criterion, Özdemir's algorithm [4] uses a weighted sum of the distances among color vectors as a searching criterion in its color-quantization process to prevent excess accumulation of quantization errors. Breaux's algorithm [18] is an extension of Peli's algorithm [11] for producing halftoned color-quantized images.

As mentioned in Section I, some conventional halftoning algorithms can be used in constrained halftoning to produce scalable color-quantized images after an extension. Riemersma's algorithm [12] and Orchard's algorithm [2] were extended with the scheme presented in Section 3 to provide corresponding outputs for comparison. These extensions are, respectively, referred to as Riemersma+ and Orchard+ in this paper.

S-CIELAB color difference ( $\Delta E$ ) metric [19] is a spatial extension of the CIELAB color difference ( $\Delta E$ ) metric [20]. It is defined as the Euclidean distance between the original color pixel and its reproduction in S-CIELAB color metric space. It is widely accepted and used for measuring color reproduction error when a continuous-tone color image is reproduced with halftoning. Table I shows the performance of various algorithms in terms of the average S-CIELAB difference ( $\Delta E$ ) value of all pixels in their color quantization outputs and their corresponding down-sampled versions. The palettes used to obtain Table I(a)–(d) is, respectively, of size 16, 32, 64, and 128. Simulation results show that the proposed algorithm can provide a better

result than the other algorithms even though constrained halftoning is applied to some of them.

Fig. 3(b)–(h) shows the processing results of different evaluated algorithms. Fig. 3(a) is the original  $256 \times 256$  24-bit full-color image for reference. The palette used to generate Fig. 3(b)–(h) is of size 32 and was obtained with Fig. 3(a) using median-cut algorithm. One can see the ripple patterns in the sky region in Fig. 3(d) and (h). These patterns are caused by directional hysteresis.

As a multiscale error diffusion algorithm, the proposed algorithm eliminates directional hysteresis, and, hence, no such artifacts can be found in Fig. 3(e) and (f). Fig. 3(e) and (f) shows, respectively, the cases when the energy term is defined with (4) and (5). It is hard to tell their difference, and, hence, (5) is recommended in this paper. Though Özdemir's [4], Breaux's [18], and Riemersma's [12] algorithms can also reduce directional hysteresis, artifacts can be found in their simulation results. In particular, pattern noise can be observed in the yellow cap in Fig. 3(b) and severe color shifts can be found between the yellow and the red caps in Fig. 3(c). As compared with Fig. 3(f), the caps in Fig. 3(g) is noisy. Though these algorithms can effectively remove directional hysteresis in their produced halftones, their color quantization performance is inferior to that of the proposed algorithm.

Fig. 3(g) and (h) shows the results of applying constrained halftoning to Riemersma's [12] and Orchard's [2] algorithms. Besides the artifacts that we mentioned earlier, one can see that, as compared with Fig. 3(a), the fine details of the peak of the yellow cap are totally missing in these figures. In contrast, these details are preserved in Fig. 3(e) and (f) to a certain extent. This shows that not all algorithms can work with constrained halftoning to provide a good color quantization result.

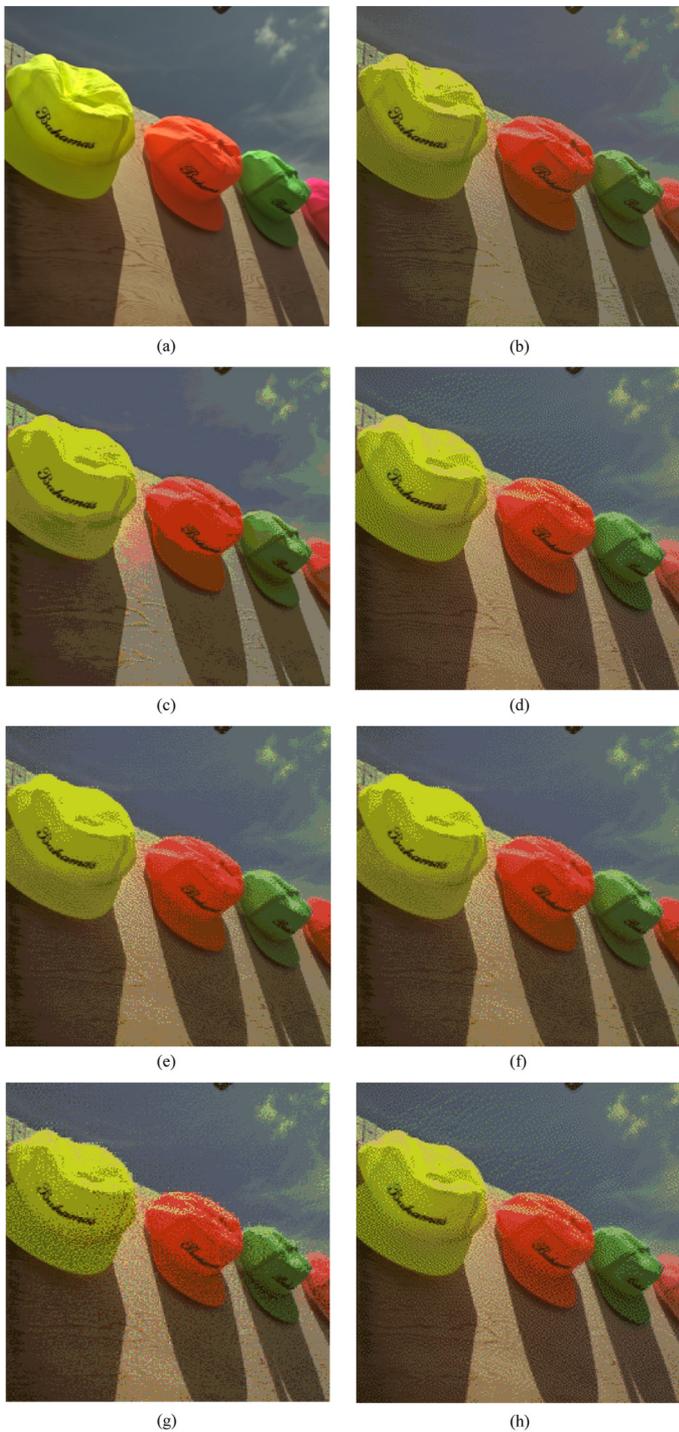


Fig. 3. Color quantization results of full-scaled “Caps” (palette size = 32): (a) Original, (b) Özdemir, (c) Breaux, (d) Akarun, (e) proposed [using energy definition (4)], (f) proposed [using energy definition (5)], (g) Riemersma+, and (h) Orchard+.

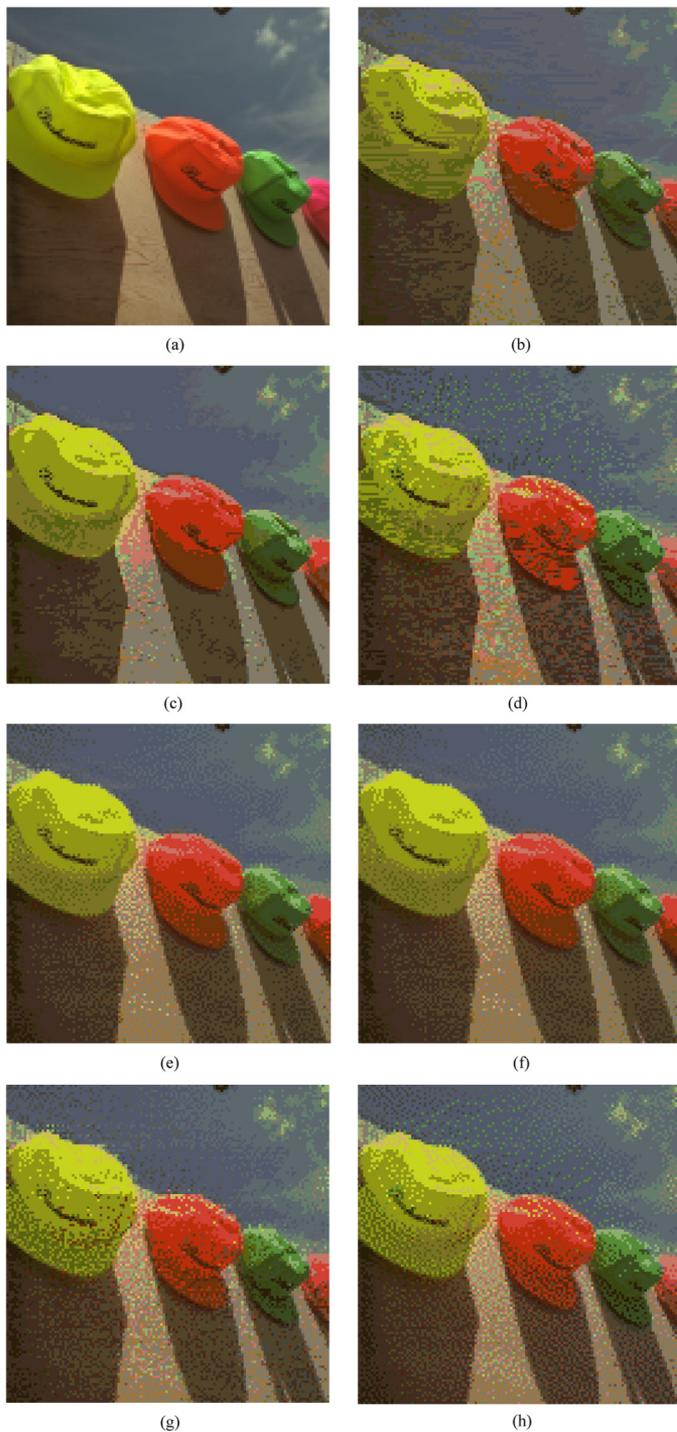


Fig. 4. Downscaled versions of Fig. 3 ( $s_r = 2$ ): (a) Original, (b) Özdemir, (c) Breaux, (d) Akarun, (e) proposed [using energy definition (4)], (f) proposed [using energy definition (5)], (g) Riemersma+, and (h) Orchard+.

Figs. 4 and 5 show the downscaled versions of Fig. 3. The downscaling of Fig. 3(b)–(h) was carried out by simple down sampling. Figs. 4(a) and 5(a) show  $I^1$  and  $I^2$ , respectively. They are used as references for evaluating how close the downscaling outputs of a color-quantization result is to the downscaled original. The corresponding downscaling ratios used to produce Figs. 4 and 5 were, respectively, 2 and 4. Accordingly, they are of size  $128 \times 128$  and  $64 \times 64$ , respectively. For easier inspection, these figures are zoomed with nearest-

neighbor interpolation to make their size as large as the original full-scaled version. One can see that the downscaled versions of Fig. 3(e) and (f) are equally good and can faithfully report the content of the downscaled versions of Fig. 3(a) while the others cannot. The poor appearance of the downscaled versions of Fig. 3(b) and (d) is expected as Özdemir’s [4] and Akarun’s [3] algorithms do not take care of the scalability of their outputs. Artifacts found in Fig. 3(c), (g), and (h) can also be found in their corresponding downscaled versions.

## V. CONCLUSION

It is always useful to produce scalable color-indexed images for delivering media information to diverse clients over heterogeneous networks reliably and efficiently. This paper presents an extension work of binary multiscale error diffusion to handle color quantization. Based on this work, a color quantization algorithm is proposed to produce scalable color-indexed images. For any given image, this algorithm can produce a high-quality directional hysteresis-free output and simultaneously embed a set of color quantization results of the downscaled versions of the given image without any memory overhead. With the color-quantization output of the proposed algorithm, images of desirable resolutions can be extracted by simply down sampling the output. The proposed algorithm works with any arbitrary color palettes of different size.

## REFERENCES

- [1] P. Heckbert, "Color image quantization for frame buffer displays," *Comput. Graph.*, vol. 16, no. 4, pp. 297–307, 1982.
- [2] M. T. Orchard and C. A. Bouman, "Color quantization of images," *IEEE Trans. Signal Process.*, vol. 39, no. 12, pp. 2677–2690, Dec. 1991.
- [3] 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, Jul. 1997.
- [4] D. Özdemir and L. Akarun, "Fuzzy error diffusion," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 683–690, Apr. 2000.
- [5] R. W. Floyd and L. Steinberg, "An adaptive algorithm for spatial grayscale," in *Proc. Soc. Information Display*, 1976, vol. 17, no. 2, pp. 75–77.
- [6] P. Stucki, MECCA—A Multiple Error Correcting Computation Algorithm for Bi-Level Image Hard Copy Reproduction IBM Res. Lab., Zurich, Switzerland, Res. Rep. RZ1060, 1981.
- [7] 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.
- [8] Y. H. Fung and Y. H. Chan, "A POCS-based restoration algorithm for restoring halftoned color-quantized images," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1985–1992, Jul. 2006.
- [9] Y. H. Fung and Y. H. Chan, "A simulated annealing restoration algorithm for restoring halftoned color-quantized images," *Signal Process.: Image Commun.*, vol. 21, no. 4, pp. 280–292, Apr. 2006.
- [10] P. W. Wong, "Adaptive error diffusion and its application in multiresolution rendering," *IEEE Trans. Image Process.*, vol. 5, no. 7, pp. 1184–1196, Jul. 1996.
- [11] E. Peli, "Multiresolution error convergence halftone algorithm," *J. Opt. Soc. Amer. A*, vol. 8, no. 4, pp. 625–633, 1991.
- [12] T. Riemersma, "A balanced dither algorithm," *C/C++ Users J.*, vol. 16, no. 12, Dec. 1998.
- [13] I. Katsavounidis and C. C. J. Kuo, "A multiscale error diffusion technique for digital halftoning," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 483–490, Mar. 1997.
- [14] Y. H. Chan, "A modified multiscale error diffusion technique for digital halftoning," *IEEE Signal Process. Lett.*, vol. 5, no. 11, pp. 277–280, Nov. 1998.
- [15] Y. H. Chan and S. M. Cheung, "Feature-preserving multiscale error diffusion for digital halftoning," *J. Electron. Imag.*, vol. 13, no. 3, pp. 639–645, 2004.
- [16] N. Damara-Venkata, B. L. Evans, and V. Monga, "Color error diffusion halftoning," *IEEE Signal Process. Mag.*, vol. 20, no. 4, pp. 51–58, Jul. 2003.
- [17] D. L. Lau, G. R. Arce, and N. C. Gallagher, "Digital color halftoning with generalized error diffusion and multichannel green-noise masks," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 923–935, May 2000.
- [18] N. Breaux and C. H. H. Chu, "Halftoning for color-indexed displays," in *Proc. Int. Conf. Image Processing*, Oct. 1999, pp. 597–601.
- [19] 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.
- [20] *CIE Recommendations on uniform color spaces, color difference equations, psychometric color terms*, Supplement No. 2 to CIE publication No. 15 (E-1.3.1) 1971/(TC-1.3.), 1978.

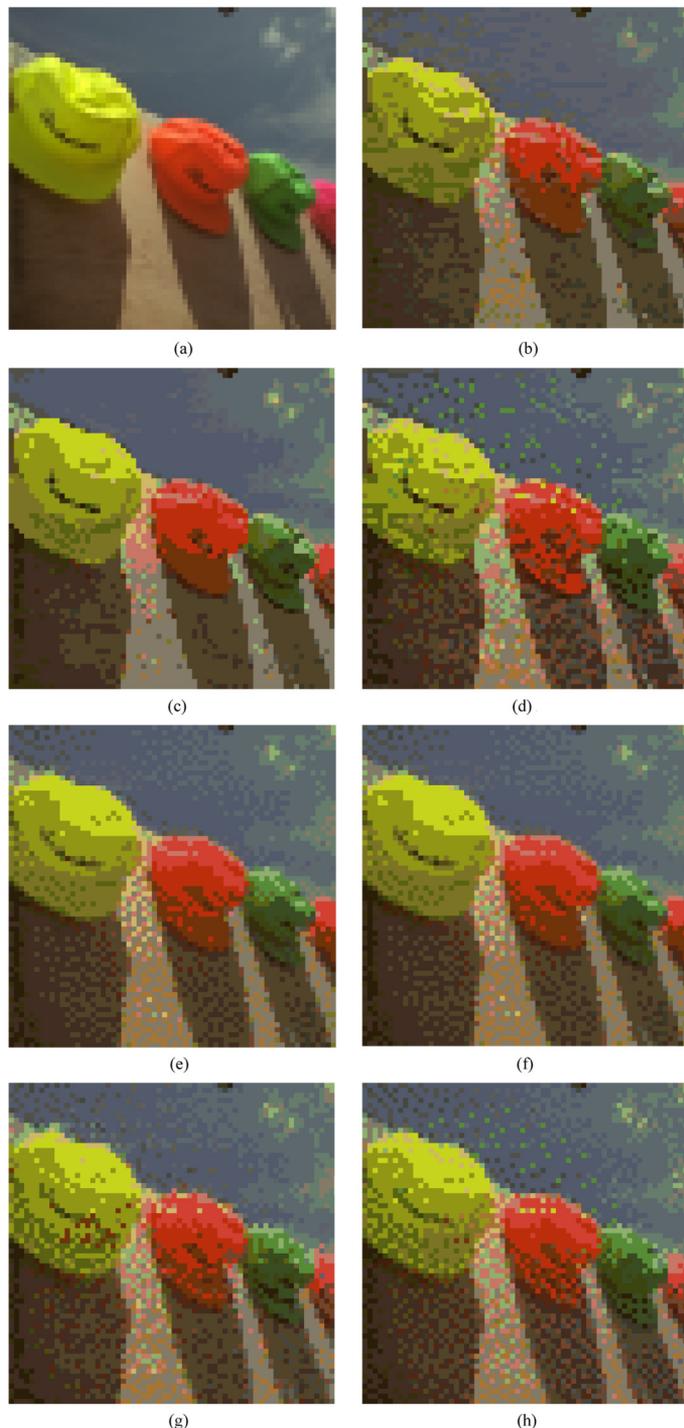


Fig. 5. Downscaled versions of Fig. 3 ( $s_r = 4$ ): (a) Original, (b) Özdemir, (c) Breaux, (d) Akarun, (e) proposed [using energy definition (4)], (f) proposed [using energy definition (5)], (g) Riemersma+, and (h) Orchard+.

Unlike the algorithms in which causal diffusion filters and predefined processing sequences are used in the error diffusion process, the proposed algorithm color-quantizes pixels in a so-called "maximum energy guidance" manner and diffuses the quantization errors with a noncausal diffusion filter. This approach completely removes the artifacts caused by directional hysteresis.