

RGBZ Image Restoration by Patch Clone

Lilong Shi^a, Iliia Ovsianikov^a, Dong-Ki Min^b, Wonjoo Kim^b, Yibing Michelle Wang^a, Grzegorz Waligorski^a, Hongyu Wang^a, Yoondong Park^b, and Chilhee Chung^c; ^aAdvanced Image Research Lab; Pasadena, CA/USA; ^bImage development team, System LSI, SEC, Yongin-si, Gyeonggi-do, Korea; ^c Samsung Advanced Institute of Technology, SEC, Yongin-si, Gyeonggi-do, Korea

Abstract

The RGBZ sensor is a novel imaging sensor that captures both color and depth images simultaneously in a single chip, with a specially designed color-filter-array (CFA), in which some of the RGB color pixels are replaced by “Z” pixels that capture depth information but no color information. As a result, RGB color images produced by this pixel array appear degraded, with missing RGB values or “holes” at locations occupied by the Z pixels. To fill in these “holes”, and thus restore resolution and appearance of color images, we propose a Patch-Clone method that exploits redundant texture information in the scene. Derived from the non-local approaches, our method consists of two steps: 1) a matching step to identify the candidate patch that contains the most useful information to reconstruct the color pixels missing at a particular hole; 2) a cloning step to copy the content from the candidate to fill in the hole. When higher order pixel content is copied, pixel continuity between the restored and original pixels can be enforced. The result of the proposed method is full resolution Bayer images, to which existing common demosaic algorithms can be applied. Tests show that the proposed method provides better reconstruction result in term of distortion error as well as visual appearance.

1. Introduction

A number of techniques to restore damaged images exist in the image processing literature. *Image inpaint* (also known as *image completion* or *disocclusion*) is an active research area to restore damaged images for plausible visual appearance. One main category of *image inpaint* approaches is geometry oriented that interpolates missing information by solving partial differential equation (PDE). The other main category is texture oriented, relying on the existence of similar structures in a given image. Both of these two categories have been reviewed in depth as variational formulations of non-local approach by Arias et al [1]. Another scenario that missing pixels need to be filled-in is *video de-interlacing*, where all odd or even rows are missing in a video frame. Recently, video *de-interlacing* methods that use edge information for video frame reconstruction have been proposed in [2-5]. The basic idea is to identify the direction of the highest correlation for directional interpolation. Compared to linear interpolation approaches, Edge-based Linear Averaging (ELA) [2] may reconstruct sharply oblique lines, avoiding many of the jagged edges produced by line doubling or line averaging. Another related problem is called *error concealment* in video transmission, where some data packets carrying image information in different frequency domain are lost during transmission. Commonly, lost information in different sub-band coded images is reconstructed separately via interpolating information of neighboring pixels [6-8].

In this paper, our work focuses on reconstructing visually plausible color images based on an experimental RGBZ sensor manufactured by Samsung Electronics Corporation [9]. The RGBZ sensor captures color and depth images simultaneously, with a specially designed color filter array (CFA) derived from the standard Bayer CFA where some RGB pixels are replaced by “Z” pixels that capture only the depth information. Because the Z pixels capture no visible color information, they appear as “holes” in an RGBZ image. An instance of the RGBZ sensor layout is shown in Figure 1, where a hole occupies an area of 4x2 color pixels, and the holes are distributed 6 pixels and 4 pixels apart in the vertical and horizontal directions, respectively. Due to the unique design of RGBZ color pattern, applying commonly used demosaicing algorithms for the standard Bayer CFA will produce black “holes” in the full color image, unless a substitute for the lost pixel information can be provided. To this end, we propose the *Patch Clone* method to fill-in the missing pixels in an RGBZ image to reconstruct a full Bayer CFA, thereby existing standard demosaic algorithms can be applied. Our approach is similar to the *Non-local* based approaches described in [1], but is more efficient for real time application. To achieve that, instead of estimating each individual missing pixel in a special order, we fill in the entire patch (ie. hole) at once. Moreover, unlike any of the techniques above, the result of our reconstruction is a full Bayer image, given an input RGBZ raw image where adjacent pixels contain signals from different color bands (or black at Z pixels).

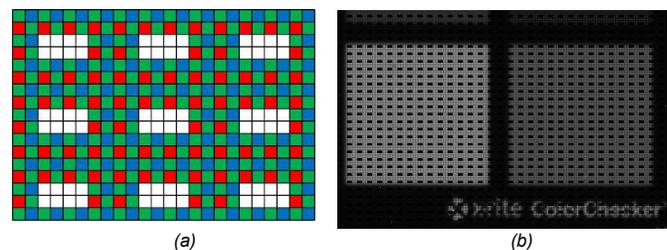


Figure 1. (a) CFA of the new type of RGBZ sensor. The red, green and blue squares represent the locations for red, green and blue color filters; the white squares represent the locations of Z sensors. (b) A real RGBZ image taken with an RGBZ sensor [9], where Z-pixels appear as black “holes”(original image is cropped for display)

In section 2, we detail our method that consists in two steps: the *matching* step and the *cloning* step. In the *matching* step, we search for the texture that best matches the “template” defined as all boundary pixels surrounding a hole. In the second step, the missing pixels in the hole are filled in by cloning those from the best match in the first step. As our method intends to produce “perceptually plausible” images, in Section 3 we test our method based on widely used evaluation criteria, namely, the Mean Square Error (MSE), the Peak Signal-to-Noise Ratio (PSNR), and

Structural Similarity (SSIM) [10], for measuring the fidelity between the reconstructed image and the reference image. Also, the performances of the proposed reconstruction are compared to bilinear interpolation and ELA-based approaches based on 12 images of the benchmark Kodak database [11]. In the end, the conclusion of our paper is provided in Section 4.

2. Method Description

2.1. Template Matching

In the first step of the *patch clone* method, we search for the texture within the image that best matches the *template*. In our case, the *template* is the patch composed of all *boundary pixels* around the target hole to be filled. A *hole* is defined as a group of pixels that are 4-connected (in terms of connectivity). A *boundary pixel* around a hole is a color (red, green or blue) pixel that is 8-connected to any pixel belonging to the hole.

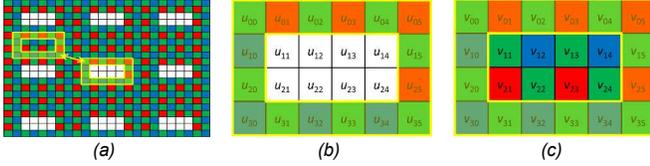


Figure 2. (a) The matching mechanism. The boundary pixels of the template and candidate patches are indicated by yellow outlines. (b) Hole and boundary pixels of the template at location \mathbf{x} , labeled by local coordinate. The yellow highlight indicates boundary pixels. (c) Pixels of the best matching candidate at location \mathbf{y} , labeled by local coordinate. The yellow highlight indicates boundary pixels.

Hence, for every hole to be filled-in, we first identify the *template* patch that consists of *boundary pixels* surrounding the hole in the data image. Then we search for a *candidate* patch that best matches the *template*. A *candidate* is a patch that is of the same size as the template and retains the same color filter arrangement. We match the *template* by sliding it within an $M \times N$ neighborhood in a way that the underlying *candidate* patch has the same color order as the *template*. The similarity of the *template* and the *candidate* is computed based on the boundary pixels only, excluding any pixel belonging to a hole. This *matching* mechanism is illustrated in Figure 2(a).

We can formulate our model in a more general mathematical form. Given \mathbf{O} denote the set of all locations of pixels belonging to the holes in an image, and let \mathbf{x} denote a set of pixel locations of a single hole. Clearly, \mathbf{x} is a subset of \mathbf{O} , i.e. $\mathbf{x} \subset \mathbf{O}$. Let $h(\mathbf{x})$ be the set of pixel values at location \mathbf{x} , and $b(\mathbf{x})$ the set of pixel values on the boundary of the hole, provided $b(\mathbf{x}) \cap h(\mathbf{O}) = \emptyset$. Therefore, the *matching* step can be derived from the general framework of the classic non-local approach, that is,

$$h(\mathbf{x}) = \sum_{s, \mathbf{x}+s \in \mathbf{O}} w(\mathbf{x}, \mathbf{x}+s) h(\mathbf{x}+s) \quad (1)$$

Here “+” is the shift operator for offset s , which defines the size of the neighborhood to search for around \mathbf{x} . The condition $(\mathbf{x}+s) \in \mathbf{O}$ explicitly excludes the case that $\mathbf{x}+s$ is in a hole, e.g. when $s = 0$. The weight function w measures the similarity between two vectors. In the classic non-local framework, Gaussian weights are commonly used,

$$w(\mathbf{x}, \mathbf{x}+s) = \exp(-\frac{1}{\sigma} \|b(\mathbf{x}) - b(\mathbf{x}+s)\|) / \sum_{s, \mathbf{x}+s \in \mathbf{O}} \exp(-\frac{1}{\sigma} \|b(\mathbf{x}) - b(\mathbf{x}+s)\|) \quad (2)$$

where $\|\cdot\|$ is the L_2 -normal of two vectors and σ defines the shape of Gaussian function. In the proposed method, we set $\sigma \rightarrow 0$, then w becomes an impulse function, i.e.

$$w(\mathbf{x}, \mathbf{x}+s) = \begin{cases} 1, & \text{if } \min \|b(\mathbf{x}) - b(\mathbf{x}+s)\|, \text{ and } \mathbf{x}+s \in \mathbf{O} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The definition of the weight function w in Equation 3 allows us to find the *best matching candidate*, at location \mathbf{y} . Optionally, the weight w can be luminance-invariant when the *boundary* pixels are normalized with respect to the luminance, that is, $\tilde{b}_i = b_i / \sum b_i$, where b_i is a pixel value of the boundary and $b_i \in b(\mathbf{x})$. Hence, w can then be expressed as

$$w(\mathbf{x}, \mathbf{x}+s) = \begin{cases} 1, & \text{if } \min \|\tilde{b}(\mathbf{x}) - \tilde{b}(\mathbf{x}+s)\|, \text{ and } \mathbf{x}+s \in \mathbf{O} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

The luminance-invariant weight function allows us to compute the difference between two patches when they differ because of texture rather than luminance.

2.2. Cloning

Once the *best matching candidate* is identified, its content can be used to fill in the hole surrounded by the *template*. The combination of *matching* and *cloning* can be expressed as the non-local framework as in Equation (1) and (3). There are three ways to clone the content from the *best matching candidate*: by pixel values (0th-order), by the pixel gradient (1st-order), and by Laplacian (2nd-order).

2.2.1 Copy by Value (0th-Order)

In this case, the non-boundary pixel values of the *best matching candidate* are directly copied to hole of the *template* (the exact Equation (1) is used). Suppose the best matching candidate is found at location \mathbf{y} , then we have

$$h(\mathbf{x}) = h(\mathbf{y}) \quad (5)$$

That is, $u_{i,j} = v_{i,j}$, for $i = 1, 2$ and $j = 1, 2, 3, 4$, where i and j are in local coordinate of a patch as illustrated in Figure 2 (b) and (c).

2.2.2 Copy by Gradient (1st-Order)

In this case, gradient of the non-boundary pixel values of the *best matching candidate* is cloned to the hole of the *template*. According to Equation (1), we have

$$\nabla h(\mathbf{x}) = \sum_{s, \mathbf{x}+s \in \mathbf{O}} w(\mathbf{x}, \mathbf{x}+s) \nabla h(\mathbf{x}+s) \quad (6)$$

or,

$$\nabla h(\mathbf{x}) = \nabla h(\mathbf{y})$$

The gradient, denoted by ∇ , is defined as the change of pixel values in vertical and horizontal directions. In general

$$\nabla u_{i,j} = \frac{(u_{i,j+1} - u_{i,j}) + (u_{i,j} - u_{i,j-1})}{2\Delta x} + \frac{(u_{i+1,j} - u_{i,j}) + (u_{i,j} - u_{i-1,j})}{2\Delta y} \quad (7)$$

$$\nabla v_{i,j} = \frac{(v_{i,j+1} - v_{i,j}) + (v_{i,j} - v_{i,j-1})}{2\Delta x} + \frac{(v_{i+1,j} - v_{i,j}) + (v_{i,j} - v_{i-1,j})}{2\Delta y} \quad (8)$$

Here Δx and Δy are both 1. Given $\nabla u_{i,j} = \nabla v_{i,j}$, and $v_{i,j}$ are known, the hole pixels, $u_{i,j}$, can be therefore calculated. However, the pixels adjacent to a non-boundary pixel are not always unknown in the vertical or horizontal direction. For example, consider the following two cases: 1) in Figure 2(b), at pixel u_{11} , pixels u_{12} and u_{21} are unknown; 2) in Figure 2(c), at pixel v_{11} , pixel v_{01} is unknown if $v_{01} \in h(\mathbf{O})$. Therefore, we need to re-define the gradient and compute the hole pixel in a case-by-case manner to ensure all involved pixels are available. Let us first define a function t that checks whether a pixel belongs to a hole, that is,

$$t_y = \begin{cases} 1, & \text{if } v_y \notin h(\mathbf{O}) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Then according to Equation (7)-(9), the values u_{11} , u_{21} , u_{14} , and u_{24} can be calculated as

$$u_{11} = (t_{01}(u_{01} - v_{01}) + t_{10}(u_{10} - v_{10})) / (t_{01} + t_{10}) + v_{11} \quad (10)$$

$$u_{21} = (t_{31}(u_{31} - v_{31}) + t_{20}(u_{20} - v_{20})) / (t_{31} + t_{20}) + v_{21}, \quad (11)$$

$$u_{14} = (t_{04}(u_{04} - v_{04}) + t_{15}(u_{15} - v_{15})) / (t_{04} + t_{15}) + v_{14}, \quad (12)$$

$$u_{24} = (t_{34}(u_{34} - v_{34}) + t_{25}(u_{25} - v_{25})) / (t_{34} + t_{25}) + v_{24}, \quad (13)$$

The calculation of u_{12} , u_{22} , u_{13} and u_{23} is more complicated,

$$u_{12} = \begin{cases} u_{02} - v_{02} + v_{12} & \text{if } t_{02} \neq 0 \\ (t_{01}(u_{01} - v_{01}) + t_{03}(u_{03} - v_{03})) / (t_{01} + t_{03}) + v_{12} & \text{else if } t_{01} \neq 0 \text{ or } t_{03} \neq 0 \\ (u_{10} - v_{10} + u_{32} - v_{32}) / 2 + v_{12} & \text{otherwise} \end{cases} \quad (14)$$

$$u_{22} = \begin{cases} u_{32} - v_{32} + v_{22} & \text{if } t_{32} \neq 0 \\ (t_{31}(u_{31} - v_{31}) + t_{33}(u_{33} - v_{33})) / (t_{31} + t_{33}) + v_{22} & \text{else if } t_{31} \neq 0 \text{ or } t_{33} \neq 0 \\ (u_{20} - v_{20} + u_{02} - v_{02}) / 2 + v_{22} & \text{otherwise} \end{cases} \quad (15)$$

$$u_{13} = \begin{cases} u_{03} - v_{03} + v_{13} & \text{if } t_{03} \neq 0 \\ (t_{02}(u_{02} - v_{02}) + t_{04}(u_{04} - v_{04})) / (t_{02} + t_{04}) + v_{13} & \text{else if } t_{02} \neq 0 \text{ or } t_{04} \neq 0 \\ (u_{15} - v_{15} + u_{33} - v_{33}) / 2 + v_{13} & \text{otherwise} \end{cases} \quad (16)$$

$$u_{23} = \begin{cases} u_{33} - v_{33} + v_{23} & \text{if } t_{33} \neq 0 \\ (t_{32}(u_{32} - v_{32}) + t_{34}(u_{34} - v_{34})) / (t_{32} + t_{34}) + v_{23} & \text{else if } t_{32} \neq 0 \text{ or } t_{34} \neq 0 \\ (u_{25} - v_{25} + u_{03} - v_{03}) / 2 + v_{23} & \text{otherwise} \end{cases} \quad (17)$$

The intuition is that we first try to use boundary pixels that are within 4-connected neighborhood to compute the gradient by Equation 8. If no such neighbor exists, we use 8-connected



Figure 3. Thumbnails of the 12 images from Kodak database [11]

neighbors. Otherwise, we use pixels that are in horizontal and vertical directions in a 5x5 neighborhood.

2.2.3. Copy by Laplacian (2nd order gradient)

In this case, Laplacian of the non-boundary pixel values of the *best matching candidate* is cloned to the hole of the *template*. According to Equation (1), we have

$$\nabla^2 h(\mathbf{x}) = \sum_{s, \mathbf{x}+s \in \mathbf{O}} w(\mathbf{x}, \mathbf{x}+s) \nabla^2 h(\mathbf{x}+s) \quad \text{or,} \quad (18)$$

$$\nabla^2 h(\mathbf{x}) = \nabla^2 h(\mathbf{y}).$$

Here ∇^2 is the discrete 2D Laplacian operator defined as

$$\nabla^2 u_{i,j} = \frac{1}{\Delta x \Delta y} (u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}) \quad (19)$$

$$\nabla^2 v_{i,j} = \frac{1}{\Delta x \Delta y} (v_{i+1,j} + v_{i-1,j} + v_{i,j+1} + v_{i,j-1} - 4v_{i,j}) \quad (20)$$

Combining Equation (18) – (20), we get

$$u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = \nabla^2 u_{i,j} = \nabla^2 v_{i,j} \quad (21)$$

Therefore, given Equation (21) and $v_{i,j}$ available, we can estimate all u 's in the hole at once by solving the following linear system of equations,

$$\begin{bmatrix} 4 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 4 & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & 4 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 4 & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & 4 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 4 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 4 \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \\ u_{13} \\ u_{23} \\ u_{14} \\ u_{24} \end{bmatrix} = \begin{bmatrix} -\nabla^2 v_{11} + u_{01} + u_{10} \\ -\nabla^2 v_{21} + u_{20} + u_{31} \\ -\nabla^2 v_{12} + u_{02} \\ -\nabla^2 v_{22} + u_{32} \\ -\nabla^2 v_{13} + u_{03} \\ -\nabla^2 v_{23} + u_{33} \\ -\nabla^2 v_{14} + u_{04} + u_{15} \\ -\nabla^2 v_{24} + u_{25} + u_{34} \end{bmatrix} \quad (22)$$

Similar to the 1st-order case, when not all pixels in the *best matching candidate* are known, the Laplacian cannot be computed by Equation (20). Thus we have to modify Equation (21) to compute the Laplacian function by using only available pixels, that is

$$\nabla^2 v_{i,j} = t_{i+1,j} v_{i+1,j} + t_{i-1,j} v_{i-1,j} + t_{i,j+1} v_{i,j+1} + t_{i,j-1} v_{i,j-1} - T_{i,j} v_{i,j} \quad (23)$$

where $T_y = t_{i+1,j} + t_{i-1,j} + t_{i,j+1} + t_{i,j-1}$. Here, t is the binary function defined in the Section 2.2.2 and T is the count of no-hole pixels

within 4-connected neighborhood. Accordingly, the linear system in Equation (22) is modified to

$$\begin{bmatrix} T_{11} & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ -1 & T_{21} & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & T_{12} & -1 & -1 & 0 & 0 & 0 \\ 0 & -1 & -1 & T_{22} & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & T_{13} & -1 & -1 & 0 \\ 0 & 0 & 0 & -1 & -1 & T_{23} & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & T_{14} & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & T_{24} \end{bmatrix} \begin{bmatrix} u_{11} \\ u_{21} \\ u_{12} \\ u_{22} \\ u_{13} \\ u_{23} \\ u_{14} \\ u_{24} \end{bmatrix} = \begin{bmatrix} -\nabla^2 v_{11} + t_{01} u_{01} + t_{10} u_{10} \\ -\nabla^2 v_{21} + t_{20} u_{20} + t_{31} u_{31} \\ -\nabla^2 v_{12} + t_{02} u_{02} \\ -\nabla^2 v_{22} + t_{32} u_{32} \\ -\nabla^2 v_{13} + t_{03} u_{03} \\ -\nabla^2 v_{23} + t_{33} u_{33} \\ -\nabla^2 v_{14} + t_{04} u_{04} + t_{15} u_{15} \\ -\nabla^2 v_{24} + t_{25} u_{25} + t_{34} u_{34} \end{bmatrix} \quad (24)$$

Again, the hole pixels surrounded by the *template* can be calculated by solving Equation 24.

In summary, the 0th-order cloning copies the pixel values to the hole directly, with no guarantee of the pixel value continuity between the estimated pixels hole and the pixels of *template*. As a result, sharp intensity change from hole to boundary pixels can cause artifacts such as white or dark spots. These artifacts may be propagated to the neighboring pixels during demosaicing [12]. On the other hand, the 1st-order cloning enforces the continuity between estimated hole pixels and the *template* pixels, but not the continuity among the pixels of the hole. The 2nd-order cloning, based on discrete Poisson equations, enforces continuity between every two adjacent pixels of the patch (including both hole and boundary pixels), and therefore minimizes the potential artifacts.

For generality, the proposed approach does not assume the shape of the holes, or their distribution in the image, or the image color pattern of the image. Just for our RGBZ sensor, a patch is a rectangle that occupies a 4x6 area and the shift s should be a multiple of 2 horizontally/vertically in order to ensure the same color order of the *template* and its match during the *matching* step.

3. Test Result

In this section we evaluate the image reconstruction quality of our method based on a widely used benchmark consists of 12 images of the Kodak database [11] and one Siemens star chart image. The thumbnails of Kodak images are provided in Figure 3. In the first test, we simulate RGBZ images by sampling the full color Kodak images according to the CFA shown in Figure 1. Then we process the simulated RGBZ images by the simple bilinear interpolation approach, the Edge-based Linear Averaging (ELA) approach [2], and the three variance of our Patch-Clone approach (with illuminant invariant). During the *matching* step, our search is within a 20x20 neighborhood. Because the shift $s = 2$, the *template* is compared to less than 100 local *candidates*. Since the resultant images are full Bayer images, the reconstruction error is first measured for Bayer images only. to perform an error evaluation corresponding to human perception of final restored images in full color, we need to apply a demosaic algorithm. Although the color image reconstruction error will also depend on the choice of demosaic algorithm, the focus of this paper is to restore missing Z pixels, and hence evaluation of the effects of using different demosaic algorithms is beyond the scope of this paper. Therefore, we choose a common demosaic algorithm by Malvar et al [12], which is implemented in Matlab as a built-in function. In the second test, we reconstruct an RGBZ star chart image by filling in the Z pixels based on the proposed approaches. The reconstructed

full color image is compared to those by using bilinear interpolation and Edge-based Linear Averaging approach [2].

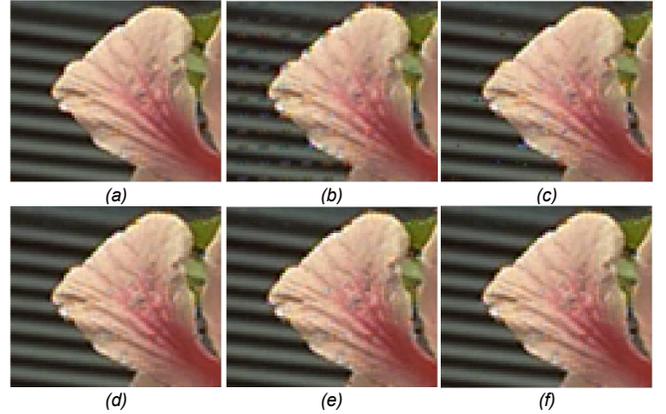


Figure 4. Image restoration comparison (original image is cropped for display). (a) reference full color image (image index 3 of the Kodak data set [11]) (b) reconstructed by bilinear interpolation (c) reconstructed by Edge-based Linear Averaging (d) 0th order Patch Clone with $M=N=20$; (e) 1st order Patch Clone with $M=N=20$; (f) 2nd order Patch Clone with $M=N=20$. All images are demosaiced by Malvar et al [12].

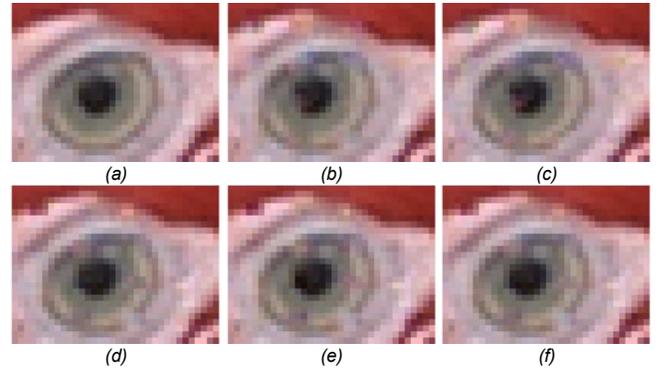


Figure 5. Image restoration comparison (original image is cropped for display) (a) reference full color image (image index 1 of the Kodak data set [11]) (b) reconstructed by bilinear interpolation (c) reconstructed by Edge-based Linear Averaging (d) 0th order Patch Clone with $M=N=20$; (e) 1st order Patch Clone with $M=N=20$; (f) 2nd order Patch Clone with $M=N=20$. All images are demosaiced by Malvar et al [12].

In the following, we provide both visual and numerical result of our methods. First of all, visual comparisons of the result are demonstrated in Figure 4-6. In Figure 4 and Figure 5, two images from the Kodak database are selected for visual comparison and the original distortion-free images are provided in Figure 4(a) and Figure 5(a) as references. The reconstructed image by bilinear interpolation and ELA are shown in (b)-(c) of Figure 4-6, where artifacts in the restoration artifacts are quite strong especially in the high frequency edge regions (shown as high contrast black/white stripes in Figure 4 and Figure 6). This is mainly because of neglecting or misidentifying the edges around the holes, which leads to wrong interpolation of the missing pixels. On the other hand, all variants of our Patch-Clone method generate visually plausible images, relying on the fact that similar texture structures usually present in nearby neighborhood. Also, visual differences

between the images produced by the three variants are quite subtle. For instance, one may find it very hard to tell the difference in the reconstructions in (d)-(f) of Figure 4 and 6. Therefore, a special case is provided in Figure 5 to illustrate the difference of our proposed methods with different orders. Here inside the pupil area of parrot, the 2nd order cloning (Figure 5(f)) is able to restore the image to the distortion-free reference (Figure 5(a)) better than all the other tested methods, by enforcing local continuity.

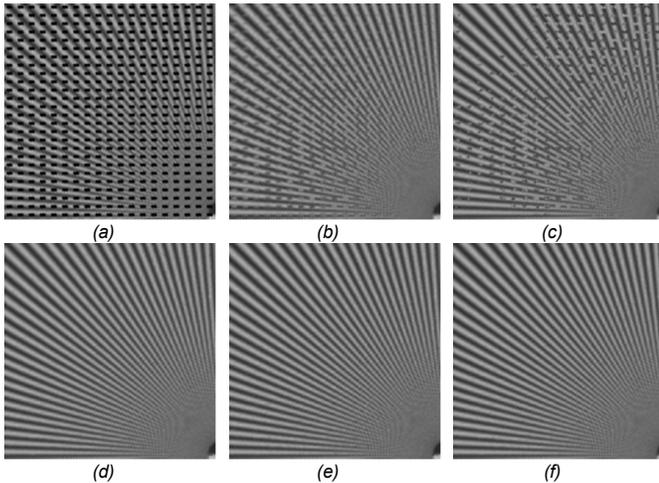


Figure 6. Comparison by star chart (original image is cropped for display) (a) full color image without Z-pixel restoration (b) reconstructed by bilinear interpolation (c) reconstructed by Edge-based Linear Averaging (d) 0th order Patch Clone with $M=N=20$; (e) 1st order Patch Clone with $M=N=20$; (f) 2nd order Patch Clone with $M=N=20$. Images are demosaiced by Malvar et al. [12]

Table 1 demonstrates that Patch-Clone based methods outperform the bilinear interpolation and ELA in all four error measures. Interestingly, the 0th order Patch Clone method has the lowest complexity yet the performance is slightly better than the 1st order method. On the other hand, the 2nd order Patch Clone shows evidently better image reconstruction quality than the other two. Please note that the MSE, PSNR and SSIM are only objective qualitative measures for image distortion, rather than a perceptual measurement by human experiments.

Table 1. Comparison of image restoration methods based on the Kodak database [11]. The quality of the image is evaluated based on MSE (of Bayer and full color images in CIE $L^*a^*b^*$ color space), PSNR, and SSIM.

Method	Bayer Recon. Eval.	Color Image Reconstruction Evaluation		
	MSE (x100)	PSNR	MSE (in $L^*a^*b^*$)	SSIM
Bilinear Interpolation	5.821	33.00	4.199	0.9731
Edge-based Linear Avg	6.223	32.99	4.564	0.9734
Patch Clone (0 th order)	4.397	34.29	2.898	0.9743
Patch Clone (1 st order)	4.436	34.25	2.987	0.9740
Patch Clone (2 nd order)	4.082	34.59	2.669	0.9752

4. Conclusion

Typical digital image data are characterized by a great deal of redundancy that can be exploited to reconstruct incomplete or damaged images. Based on non-local similarity, the proposed method restores the missing pixels in RGBZ images. The result is full Bayer images retaining the original image resolution. Our approach searches for the best match of the *template* composed of pixels around a hole resulting from replacement of some color pixels by a typically larger Z pixel. Once the best match is identified in the local neighborhood, the missing information is cloned from the best match, assuming the similar structure is within the neighborhood. Unlike the interpolation based approaches that may produce artifacts (such as blurring edges), which reduces the image resolution due to erroneous detection in a non-dominant directional edge regions, our method takes a different approach than local interpolation. The 0th order Patch Clone method reconstructs the image efficiently. Moreover, the 1st and 2nd order cloning reduces artifacts by enforcing local pixel continuity. Although the Patch Clone method is proposed for images from RGBZ sensors, the scheme can be easily modified to repair other damaged or incomplete images or videos.

References

- [1] P. Arias, V. Caselles, G. Facciolo, and G. Sapiro, "Variational framework for exemplar-based image inpainting," *Int. J. Comput. Vis.*, 93(3), pg. 1-29 (2011)
- [2] T. Doyle, "Interlaced to sequential conversion for EDTV applications," In 2nd. Int. Workshop Signal Processing of HDTV, 412-430 (1998).
- [3] H. Y. Lee, J. W. Park, T. M. Bae, S. U. Choi, and Y. H. Ha, "Adaptive Scan Rate Up-Conversion System Based on Human Visual Characteristics," *IEEE Transactions on Consumer Electronics* 46(4), pg. 999-1006 (2000)
- [4] Y. L. Chang, S. F. Lin, C. Y. Chen, and L. G. Chen, "Video de-interlacing by adaptive 4-field global/local motion compensated approach," *IEEE Trans. Circuits Syst. Video Techn* 15(12), pg. 1569-1582 (2005).
- [5] C. C. Lin, M. H. Sheu, H. K. Chiang, and C. Liaw, "Motion adaptive de-interlacing with horizontal and vertical motions detection," In *Proc of the 6th Pacific-Rim conference on Advances in Multimedia Information Processing - Volume Part I (PCM'05)*, pg. 291-302 (2005).
- [6] I. Bajic and J. Woods, "Domain-based multiple description coding of images and video," *IEEE Trans. Image Process.*, vol. 12 no. 10, pg. 1211-1225, (2003).
- [7] I. Bajic, "Adaptive MAP error concealment for depressively packetized wavelet-code images," *IEEE Trans. Image Process.*, vol. 15, no. 5, pg. 1226-1235, (2006).
- [8] S. Hemami and R. Gray, "Subband-coded image reconstruction for lossy packet networks," *IEEE Trans. Image Process.*, vol. 6, no. 4, pg. 523-539 (1997).
- [9] W. Kim, W. Wang, I. Ovsianikov, S. H. Lee, Y. Park, C. Chung, and E. Fossum, "A 1.5Mpixel RGBZ CMOS image sensor for simultaneous color and range image capture," 2012 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pg. 392-394 (2012).
- [10] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pg. 600-612 (2004).

- [11] Eastman Kodak and various photographers, 1991. Kodak Photo CD PCD0992, Access Software & Photo Sampler, Final version 2.0. [*CD-ROM, Part No. 15-1132-01*].
- [12] H. S. Malvar, L. W. He, R. Cutler, "High-quality linear interpolation for demosaicing of Bayer-patterned color images," in Proc. of IEEE International Conference of Acoustic, Speech and Signal Processing (ICASSP), (2004)