# Color Filter Array Demosaicking Using Self-validation Framework

Ting-Chun Wang[1], Yi-Nung Liu[2] and Shao-Yi Chien[2]
[1]Department of Electrical Engineering, [2]Graduate Institute of Electronics Engineering
National Taiwan University
1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan
[1]b97901116@ntu.edu.tw, [2]{yinung, sychien}@media.ee.ntu.edu.tw

*Abstract*—**Color demosaicking is well known as an ill-posed problem of sensor image restoration. In this paper, a self-validation framework for color demosaicking is proposed. In the proposed self-validation framework, multiple algorithms under different hypotheses will be performed to generate multiple candidates. Then the final estimation of a missing color sample will be decided by evaluating the local consistency of each algorithm with double interpolation. With this framework, the strengths of different algorithms can be combined and thus eliminate color artifacts. Experimental results demonstrate that the proposed framework can improve the image quality in both subjective and objective measures.**

***Index Terms— Demosaicking, self-validation, double interpolation, zipper effect***

## I. INTRODUCTION

In modern CMOS sensors in digital cameras, a color filter array (CFA) is widely used to capture images. Due to the concern of cost and complexity, the most popular color filter array among all different types of CFA is the Bayer pattern [1]. However, since only one color channel is captured at each pixel, the other two missing color components need to be estimated from pixels around. This process is known as demosaicking.

A lot of demosaicking algorithms have been developed to solve this problem [2]. Probably the most straightforward one is the bilinear interpolation, which simply averages the nearby pixels to estimate the missing pixels. Nevertheless, the resultant images cannot get rid of the visual artifacts, often called the zipper effect (ZE) [3]. The edge-sensing interpolation calculates horizontal and vertical gradients in the G channel and chooses the result of the direction with smaller gradient. The effective color interpolation (ECI) proposed by Pei and Tam [4] assumed that the difference of colors varies slowly and used a bilinear interpolation in the color difference domain. Li's successive approximation (SA) [5] used an iterative method to update the image quality. Chen and Chien [6] concerned about hardware cost and proposed a chrominance variance-based method. The heterogeneity-projection hard-decision (HPHD) interpolation proposed by Tsai and Song [7] tried to estimate the optimal interpolation direction. Chung *et al.* [8] gave a brand-new approach to extract gradient information combined with a novel adaptive heterogeneity-projection. Chuang *et al.* [9] utilized joint bilateral filter to exploit the correlation between color channels while preserving edges. Leung *et al.* [10] used a least-squares luma-chroma demultiplexing to get excellent tradeoff between speed and quality. Horé and Ziou [11] defined a new spectral interpolation model and applied an edge-sensing model to make an adaptive algorithm. A common fact that exists in nearly all algorithms is that they all base on some assumptions or are focusing on specific cases. As a result, they may work well for certain situations. Nevertheless, when it comes to some different cases, the result may not be so pleasing. In other words, there is no one-for-all solution. However, if there is a no-reference evaluation method that could judge each algorithm when given an input without knowing the ground truth, one can gain insight of how well each algorithm performs in the current situation. As a result, the algorithm that is more likely to outperform the others could be selected. This is the thought of self-validation, which would be introduced in Section II.

After demosaicking, coming up with an approach to measure the image quality is important. This is commonly done by calculating the peak signal-to-noise ratio (PSNR). However, the PSNR alone cannot truly reflect the subjective view of human eyes. Consequently, Lu *et al.* [12] proposed a way to calculate the percentage of pixels with the zipper effect in demosaicked images. Since it can reflect the perceptual appearance better, it is a much more significant factor in measuring the quality of the images.

In this paper, the double interpolation process proposed in [13] is adopted as a means to self-validate each demosaicking algorithm. Then, for each pixel, use the validation result to choose the algorithm that best fits it. By doing this, an adaptive approach that takes different pixel conditions into account and thus makes use of different algorithms could be achieved.

This paper is organized as follows. Section II introduces the concept of self-validation framework and the double interpolation (DI) difference map. Next, the implementation is described in Section III. Section IV shows the results and comparison with other methods. Finally, Section V is the conclusion of this paper.

## II. Self-Validation

Different demosaicking algorithms have their own assumptions. In smooth regions, all algorithms may have good results. However, in some cases such as high frequency structure or texture regions, the performance of different algorithms will depend on whether their assumptions are true or not in this area. If there is an algorithm that can take advantage of all of them, it should be able to handle much more situations. This can be achieved by the self-validation process. For a given pixel condition, first perform self-validation on each algorithm, and then choose the algorithm that has the least error in the process. Performing well in the self-validation test means it also has a good chance of doing a good job in the real situation. As a result, by adaptively choosing different algorithms according to the current condition, one can combine all the advantages of these algorithms and handle all possible situations. It turns out that how to do the self-validation becomes important, which is going to be introduced next.

In the self-validation process, a method called "double interpolation [13]" is used. The process flow is shown in Fig. 1. Suppose now a Bayer pattern image $I_{Bayer}$ and a demosaicking algorithm are given. First, the input image $I_{Bayer}$ will be demosaicked with the given algorithm, and a full RGB image $I_{RGB}$ will be derived. Then, $I_{RGB}$ will be down-sampled to get two Bayer pattern-like images, $I_{Bayer\text{-}I}$ and $I_{Bayer\text{-}II}$, whose patterns are similar to Bayer's but with their G channel positions and R and B channel positions reversed compared to the original $I_{Bayer}$; i.e., the positions of G pixels on $I_{Bayer}$ are replaced with R and B pixels and the positions of R and B pixels on $I_{Bayer}$ are replaced with G pixels. The difference between $I_{Bayer\text{-}I}$ and $I_{Bayer\text{-}II}$ is that their R and B channels are reversed. Next, perform the same demosaicking process again on both $I_{Bayer\text{-}I}$ and $I_{Bayer\text{-}II}$ to get two full RGB images $I_{RGB\text{-}I}$ and $I_{RGB\text{-}II}$. Finally, down-sample the two images, to the same pattern as the Bayer one, and obtain $I_{Bayer\text{-}I'}$ and $I_{Bayer\text{-}II'}$. Compare $I_{Bayer\text{-}I'}$ and $I_{Bayer\text{-}II'}$ with the original image $I_{Bayer}$, and for every pixel, sum the two squares of differences between the original image and the obtained images. This constructs the double-interpolation difference map (DI-map). The values in the DI-map reflect how well this algorithm works on the given image since they shed light on whether this algorithm is self-consistent; if they are large, it means that this algorithm is not performing well in this region, and vice versa. By inspecting the magnitude of the values in the DI-maps, one can gain insight into which algorithm to pick to demosaick the given pixels.

## III. Implementation

Suppose now a pool of algorithms could be utilized. With the self-validation process in section II, each algorithm could be examined and the best algorithm for each pixel could be decided independently, thus forming an adaptive algorithm.

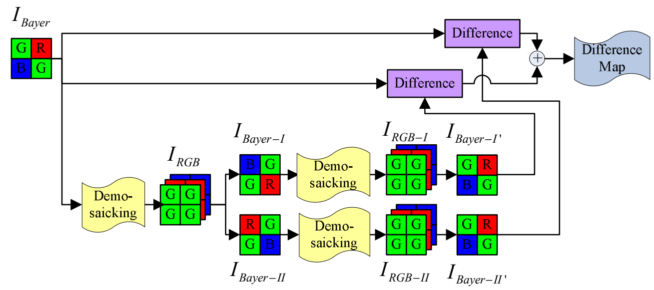



Figure 1. Flow diagram of double interpolation.

In our real work, two algorithms were selected as candidate methods: JBF [9] and Chen's [6] algorithms. The JBF's algorithm uses joint bilateral filter to exploit the correlation between color channels with the initialized information while preserving edges. Chen's method utilizes the chrominance variance weighting scheme to give a cost effective method. These two algorithms were chosen because they focus on different aspects: Chen's algorithm is good at obtaining excellent PSNR, while JBF's method excels in generating very low zipper effect. As a result, they should be able to complement each other well and give better result to both the PSNR and the zipper effect. The horizontal and vertical interpolation methods were also taken in as input algorithms. The block diagram is shown in Fig. 2. More algorithms could also be exploited and theoretically, the more input algorithms, the better the result should be.

For each input image, first one algorithm will be applied to generate the demosaicked result as a candidate. Then the self-validation process mentioned in section II will be performed to generate the DI-map for the chosen algorithm. Next, for each pixel, all the values in a window neighboring the current pixel in the DI-map will be summed, which states as the "cost" for choosing this algorithm. In the summing process, a bilateral filter is applied to give different weightings on different pixels according to their spatial and color distances from the current one. Keep doing this for all the input algorithms. Finally, the result of the algorithm that generates the least difference sum will be chosen as the final result. In other words, for a pixel $p$, its best fitting algorithm index $i$ and its pixel value $I_p$ are calculated as

$$i = \arg\min_{j} \left\{ \frac{1}{w_{jp}} \sum_{q\in\Omega} M_{jq} f\left(\left\| p-q \right\|\right) g\left(\left| I_{jp} - I_{jq} \right|\right) \right\}, \quad j = 0,1,2,3 \quad (1)$$

$$I_p = I_{ip}, \quad (2)$$

where $M_{jq}$ is the $q$-th pixel value on the $j$-th DI-map, $I_{jp}$ and $I_{jq}$ are the values of pixel $p$ and $q$ on the demosaicked image done by the $j$-th algorithm, $\Omega$ denotes the spatial window, and $w_{jp}$ denotes the normalization factor. $f$ and $g$ are the Gaussians on the spatial distance and range difference,

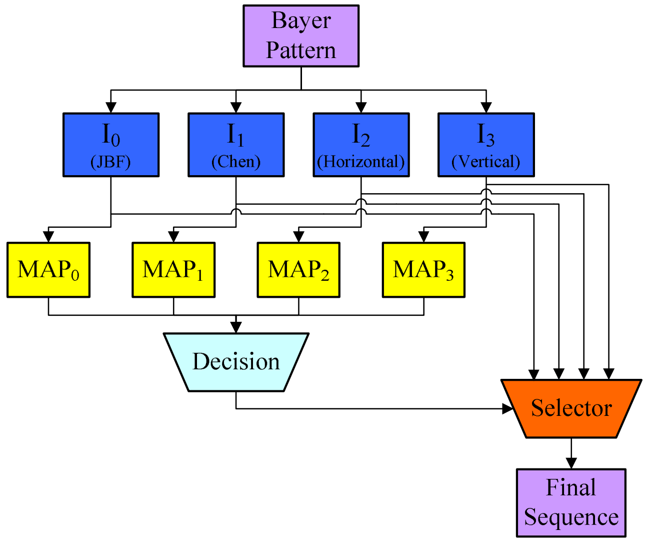



Figure 2. Block diagram of the proposed self-validation framework.

$$f\left(\|p-q\|\right)=\frac{1}{\sqrt{2\pi\sigma_s^{\,2}}}\exp\left(-\|p-q\|^2/2\sigma_s^{\,2}\right), \quad (3)$$

$$g\left(\left|I_p-I_q\right|\right)=\frac{1}{\sqrt{2\pi\sigma_r^2}}\exp\left(-\sum_{c=1}^{3}\left|I_{pc}-I_{qc}\right|^2/2\sigma_r^{\,2}\right), \quad (4)$$

where $c$ represents the R/G/B channels, $\sigma_s$ denotes the standard deviation in the spatial domain and $\sigma_r$ denotes the standard deviation in the range domain.

## IV. RESULTS

For the bilateral filter we use, the spatial standard deviation is set to 4, and the range standard deviation is set to around 0.47 with pixel values normalized to interval [0, 1]. It can be seen that the standard deviations are quite large that the Gaussians here do not play an important role. In our experimental results, the PSNRs of the results change only slightly when we tune the spatial standard deviation and the range standard deviation of the bilateral filter. Actually, the average PSNRs only decrease less than 0.3 dB for each of the three RGB channels when we remove the bilateral filter. This is a desired result since it shows that the proposed method is free of parameter constraint. For the window size, again it does not matter too much when it comes to the output PSNRs, and it was set to 25×25. The experiment showed that the larger the size, the better zipper effect we could get, but the PSNRs would decrease, so it is a tradeoff; one can easily get better PSNRs by reducing the window size while sacrificing the zipper effect, although the value ranges are very small.

Fig. 3 (a) and (b) show one of the demosaicked images of Chen's algorithm and its corresponding difference map, and (c) and (d) show the images obtained by JBF's algorithm. Note that the artifact region is correctly detected by the self-validation process, as the difference map shows. Fig. 4 (a) and (b) show the algorithm-selection mode map and the final output of the proposed algorithm. For the mode map, different colors mean different algorithms were selected: The blue region represents the JBF, green region the Chen's, and black and white regions represent the horizontal and vertical interpolations, respectively. It can be seen that the proposed algorithm made the right choice when choosing between these algorithms, demosaicking the fence area by vertical interpolation while the shutter area by horizontal interpolation. The rest of the image is not direction-dependent, so JBF and Chen's algorithms dominated.

To evaluate the performance of the proposed algorithm, the demosaicked results of 24 Kodak images are compared with some other famous algorithms, including the two inputs in the proposed algorithm. The performance of each algorithm is measured both in terms of the PSNR values of R/G/B channels respectively and the percentage of pixels with zipper effect (ZE) to their neighbors. The ZE is calculated in CIELAB color space as follows. For each pixel $p$, let $N_p$ denote the set of its eight neighboring pixels, $\Delta E_{ori}(p,q)$ and $\Delta E_{de}(p,q)$ denote the CIELAB color space distances between pixels $p$ and $q$ in the original and the demosaicked images, respectively. Then, the most similar neighbor $i_s$ and the zipper effect index $\psi$ are derived as

$$i_s=\arg\min_{i\in N_p}\Delta E_{ori}(p,i), \quad (5)$$

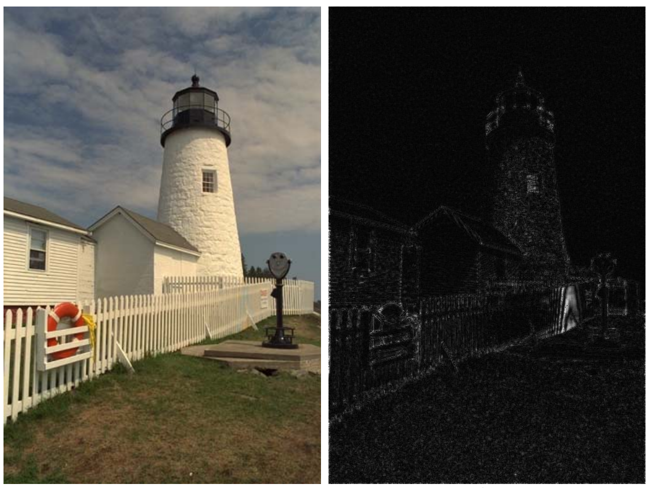
$$\psi=\Delta E_{de}(p,i_s)-\Delta E_{ori}(p,i_s), \quad (6)$$

if $|\psi|>2.3$, the zipper effect exists at pixel $p$.

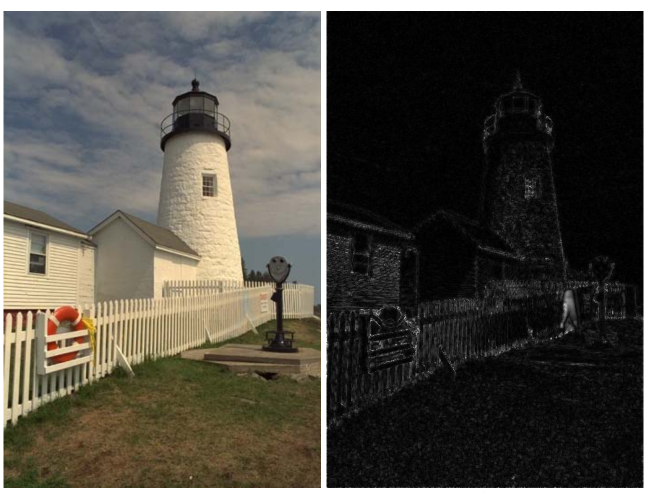
The objective evaluation results are shown in Table I and II. For the PSNRs, the proposed algorithm outperforms the others in more than half the test images. Note that, while there are lots of irregular nature textures in the input images such as grassland or water, the PSNR values of the proposed framework will be slightly lower than those of Chen's algorithm. However, the subjective quality will still be the same or even better since human perception is not able to perceive the exact pixel values of such kind of textures. The proposed framework was designed to keep the local homogeneity but not to achieve the highest PSNR. Moreover, the PSNRs of the proposed algorithm are still very close to the best one for those images. Also, in average, our proposed framework obtains the best PSNRs for all R/G/B channels as well as the ZE.

Fig. 5 and Fig. 6 show parts of the Kodak images 19 and 1 respectively. In Fig. 5, the proposed framework detected the non-homogeneity of Chen's algorithm and JBF by the DI-maps shown in Fig. 3 (b) and Fig. 3 (d), and finally chose the simple vertical filter to interpolate the fence. In Fig. 6, it is very challenging because there are both vertical and horizontal edges in the same area. The proposed framework can make decision of each pixel to make sure that there will not be unpleasing color artifacts. It can be seen that in addition to better PSNRs and ZE, the proposed algorithm also generates fewer artifacts than the others so the visual effect is better, which is much more important.

Beside the Kodak dataset, the proposed method was also applied on some latest test images like the McMaster dataset [14]. Although the proposed method is not comparable to the two methods proposed in [14], it is not the issue here; we are just trying to, by combining the input algorithms together, get a better result than those generated by them, not to beat all the state-of-the-art algorithms. It can be demonstrated that after taking in those two algorithms as inputs, the resultant method outperforms the two in the output PSNRs.



(a) (b)

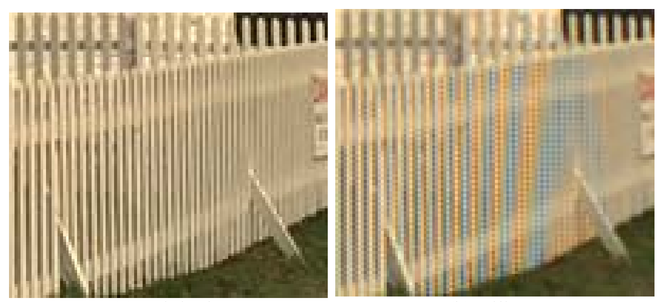


(c) (d)

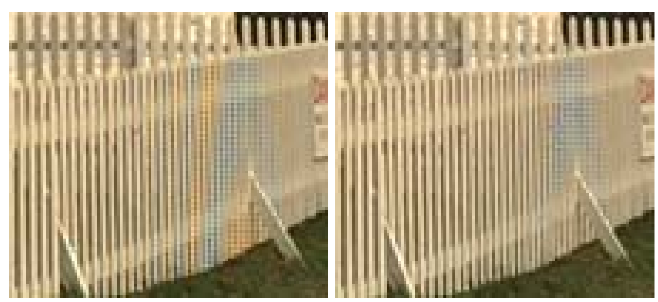
Figure 3. (a) Demosaicked image of Kodim19 using Chen's algorithm. (b) The difference map between input Bayer image and self-validation result image. (The magnitude of each pixel is four times of its original value for visual effect because the original values are too small.) (c) Demosaicked image of Kodim19 using JBF's algorithm. (d) The corresponding difference map of JBF's algorithm.

(a) (b)

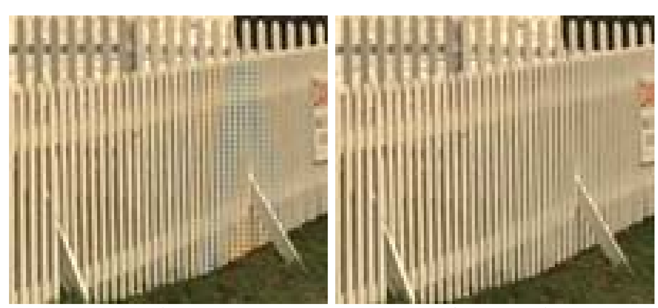
Figure 4. (a) The mode map for the proposed algorithm. Blue represents JBF, green represents Chen's, and black and white represent horizontal and vertical interpolations, respectively. (b) Demosaicked image of Kodim19 using proposed algorithm.



(a) Original (b) Bilinear



(c) ECI (d) JBF



(e) Chen's (f) Proposed

Figure 5. Comparison of different demosaicking methods for Kodim19.

TABLE I. COLOR PSNR (DB) OF 24 DEMOSAICKED IMAGES

| Img. | Bilinear | ECI | JBF | KA | Chen's | Proposed |
|---|---|---|---|---|---|---|
| 1 | 25.40 | 35.04 | 36.91 | 36.28 | 37.13 | **38.28** |
| | 29.61 | 36.64 | 40.45 | 39.12 | 40.09 | **41.34** |
| | 25.18 | 35.20 | 37.76 | 37.00 | 37.58 | **38.68** |
| 2 | 31.98 | **38.09** | 36.10 | 36.71 | 37.05 | 37.16 |
| | 36.36 | 42.51 | 42.58 | 42.25 | 43.08 | **43.55** |
| | 32.42 | 40.93 | 40.52 | 40.81 | 41.16 | **41.37** |
| 3 | 33.43 | 41.57 | 41.00 | 40.63 | 41.74 | **42.34** |
| | 37.21 | 43.97 | 43.96 | 42.82 | 45.59 | **45.89** |
| | 33.80 | 40.90 | 40.69 | 40.66 | **41.62** | 41.61 |
| 4 | 32.84 | **38.43** | 37.20 | 37.45 | 37.98 | 37.68 |
| | 36.64 | 42.26 | 42.08 | 42.23 | **43.31** | 42.86 |
| | 32.94 | 41.29 | 41.64 | 41.79 | **42.38** | 42.11 |
| 5 | 25.85 | 36.56 | 36.29 | 36.31 | 37.57 | **37.67** |
| | 29.39 | 37.94 | 39.32 | 37.78 | **41.01** | 40.93 |
| | 25.91 | 36.03 | 35.90 | 35.64 | **36.87** | 36.66 |
| 6 | 26.61 | 35.98 | 38.32 | 37.64 | 38.98 | **40.30** |
| | 31.07 | 37.80 | 41.15 | 40.32 | 41.66 | **43.04** |
| | 27.03 | 35.53 | 37.38 | 37.04 | 37.98 | **38.68** |
| 7 | 32.62 | **41.92** | 40.56 | 40.71 | 41.61 | 41.73 |
| | 36.53 | 44.08 | 43.35 | 42.74 | **45.05** | 44.99 |
| | 32.59 | **41.23** | 39.61 | 40.07 | 40.71 | 40.61 |
| 8 | 22.58 | 32.94 | 33.94 | 33.81 | 34.83 | **35.78** |
| | 27.47 | 35.26 | 37.62 | 37.16 | 38.26 | **39.58** |
| | 22.57 | 32.89 | 34.32 | 34.17 | 35.06 | **36.08** |
| 9 | 31.52 | 41.23 | 41.41 | 41.06 | 42.30 | **42.45** |
| | 35.75 | 43.19 | 44.46 | 43.03 | **45.39** | 45.27 |
| | 31.67 | 40.63 | 40.20 | 40.17 | 41.52 | **41.58** |
| 10 | 31.72 | 41.34 | 41.19 | 41.17 | **41.84** | 41.77 |
| | 35.45 | 43.47 | 44.42 | 43.86 | **45.50** | 44.99 |
| | 31.52 | 40.51 | 40.35 | 40.55 | **41.12** | 40.90 |
| 11 | 28.27 | 37.12 | 37.46 | 37.84 | 38.41 | **38.63** |
| | 32.33 | 39.05 | 41.45 | 40.55 | 42.06 | **42.54** |
| | 28.44 | 37.75 | 38.95 | 38.82 | 39.79 | **39.91** |
| 12 | 32.14 | 40.77 | 41.24 | 40.35 | 41.65 | **42.22** |
| | 36.63 | 43.73 | 44.99 | 43.55 | 45.83 | **46.23** |
| | 32.24 | 40.93 | 41.65 | 41.10 | 42.31 | **42.80** |
| 13 | 23.14 | 31.83 | 34.30 | 34.24 | 35.17 | **35.73** |
| | 26.58 | 32.58 | 36.15 | 35.51 | 36.80 | **37.69** |
| | 23.09 | 31.13 | 32.95 | 32.90 | 33.78 | **34.09** |
| 14 | 28.24 | **36.20** | 34.79 | 34.86 | 35.20 | 35.97 |
| | 32.10 | 39.06 | 38.96 | 37.95 | 39.91 | **40.55** |
| | 28.66 | **36.58** | 35.58 | 35.76 | 36.24 | 36.53 |
| 15 | 30.96 | 36.76 | 35.97 | 35.85 | **36.79** | 36.27 |
| | 34.94 | 41.27 | 41.68 | 41.10 | **43.17** | 42.24 |
| | 30.64 | 39.21 | 39.70 | 39.45 | **40.73** | 40.34 |
| 16 | 30.41 | 39.03 | 42.38 | 40.66 | 42.26 | **43.86** |
| | 34.83 | 41.16 | 45.27 | 43.61 | 45.00 | **46.53** |
| | 30.45 | 38.87 | 41.37 | 40.32 | 41.64 | **42.54** |
| 17 | 31.58 | 40.63 | 40.93 | 41.41 | **42.03** | 41.61 |
| | 34.71 | 41.26 | 43.07 | 42.11 | **44.14** | 44.04 |
| | 31.11 | 39.18 | 39.44 | 39.49 | **40.52** | 40.20 |
| 18 | 27.35 | 35.76 | 35.81 | 36.52 | **37.07** | 36.60 |
| | 30.63 | 36.54 | 38.25 | 38.10 | **39.55** | 39.49 |
| | 27.11 | 34.68 | 34.94 | 35.30 | **36.08** | 36.00 |
| 19 | 27.08 | 38.28 | 39.39 | 39.41 | 39.97 | **40.81** |
| | 31.82 | 40.03 | 42.32 | 41.74 | 42.70 | **43.81** |
| | 27.21 | 38.01 | 39.05 | 38.93 | 39.43 | **40.24** |
| 20 | 30.90 | 40.49 | 40.92 | 41.20 | 41.47 | **41.47** |
| | 34.66 | 41.70 | 43.30 | 42.93 | 43.93 | **44.07** |
| | 30.68 | 38.71 | 38.81 | 39.17 | **39.41** | 39.40 |
| 21 | 27.73 | 37.04 | 38.60 | 38.88 | 39.09 | **39.38** |
| | 31.64 | 38.33 | 41.40 | 41.04 | 41.48 | **42.08** |
| | 27.63 | 36.21 | 37.50 | 37.56 | 37.89 | **37.89** |
| 22 | 29.97 | 37.43 | 37.11 | 36.96 | **37.55** | 37.08 |
| | 33.46 | 39.57 | 39.84 | 39.49 | **40.47** | 40.21 |
| | 29.34 | 37.07 | 36.68 | 36.82 | **37.17** | 36.95 |
| 23 | 34.57 | **42.04** | 40.38 | 39.77 | 41.34 | 41.58 |
| | 38.12 | 44.69 | 44.15 | 43.06 | 45.66 | **45.67** |
| | 34.27 | 42.48 | 40.88 | 40.34 | 42.34 | **42.57** |
| 24 | 26.48 | 34.21 | 34.19 | 34.75 | **35.11** | 34.34 |
| | 29.47 | 35.30 | 36.50 | 37.01 | **37.98** | 37.64 |
| | 25.38 | 32.08 | 32.38 | 32.99 | **33.40** | 33.02 |
| Avg. | 29.31 | 37.95 | 38.18 | 38.10 | 38.92 | **39.20** |
| | 33.22 | 40.06 | 41.53 | 40.79 | 42.40 | **42.72** |
| | 29.25 | 37.83 | 38.26 | 38.20 | 39.03 | **39.20** |

(a) Original (b) Bilinear

(c) ECI (d) JBF

(e) Chen's (f) Proposed

Figure 6. Comparison of different demosaicking methods for Kodim01. There are both vertical and horizontal edges in the same area, so the directional interpolation may choose the wrong direction. Therefore, there are serious color artifacts in (b) (c) (e). The JBF method (d) put color channel into consideration and derived better result than the previous three. However, the proposed framework (f) can make decision of each pixel by checking the consistency to make sure that there is no unpleasing color artifact.

TABLE II. PERCENTAGE (%) OF PIXELS WITH ZIPPER EFFECT (ZE) TO THEIR NEIGHBORS IN DEMOSAICKED IMAGES

| Img. | Bilinear | ECI | JBF | KA | Chen's | Proposed |
|---|---|---|---|---|---|---|
| **1** | 68.31 | 19.48 | **6.08** | 9.72 | 9.35 | 6.48 |
| **2** | 27.01 | 5.18 | 4.60 | 5.12 | 5.06 | **4.51** |
| **3** | 22.11 | 4.17 | 2.28 | 2.84 | 2.48 | **2.23** |
| **4** | 32.24 | 6.05 | **4.51** | 4.89 | 4.51 | 4.77 |
| **5** | 58.30 | 16.21 | 11.21 | 12.50 | 11.48 | **10.67** |
| **6** | 52.33 | 12.04 | 4.43 | 6.19 | 5.03 | **3.38** |
| **7** | 23.25 | **4.01** | 4.09 | 4.31 | 4.08 | 4.02 |
| **8** | 68.34 | 17.52 | 10.34 | 13.72 | 12.08 | **9.18** |
| **9** | 27.19 | 3.28 | 1.62 | 2.30 | 1.93 | **1.45** |
| **10** | 28.72 | 3.10 | 1.80 | 1.99 | 2.03 | **1.78** |
| **11** | 47.12 | 9.71 | 5.76 | 6.31 | 6.06 | **5.67** |
| **12** | 28.22 | 3.13 | 1.49 | 2.44 | 1.86 | **1.45** |
| **13** | 77.39 | 31.06 | 14.11 | 16.15 | 16.33 | **13.96** |
| **14** | 56.46 | 12.77 | 8.06 | 9.27 | 9.06 | **7.92** |
| **15** | 29.85 | 8.13 | 6.09 | 7.31 | **5.99** | 6.01 |
| **16** | 40.42 | 7.29 | 1.57 | 3.05 | 2.25 | **1.13** |
| **17** | 31.91 | 5.74 | 3.19 | 3.35 | 3.28 | **3.16** |
| **18** | 53.74 | 15.58 | 10.95 | 11.38 | **10.51** | 10.82 |
| **19** | 43.48 | 8.99 | 3.16 | 3.90 | 3.91 | **2.71** |
| **20** | 26.41 | 6.25 | **3.07** | 3.34 | 3.78 | 3.15 |
| **21** | 42.12 | 11.87 | **5.15** | 5.81 | 6.45 | 5.34 |
| **22** | 40.36 | 8.68 | **5.29** | 5.97 | 5.76 | 6.07 |
| **23** | 13.62 | 2.13 | 2.11 | 2.90 | 2.09 | **1.85** |
| **24** | 47.30 | 11.92 | **7.86** | 8.18 | 8.35 | 7.90 |
| **Avg.** | 41.09 | 9.76 | 5.37 | 6.37 | 5.99 | **5.23** |

## V. CONCLUSION

In this paper, a self-validation framework is proposed to solve the color demosaicking problem. In previous works, some assumptions about the original images were made and the demosaicking algorithms were developed under those assumptions. Thus, the performances depend on whether the input images satisfy the assumptions or not. The proposed approach can make use of different demosaicking algorithms and take advantage of them by evaluating their performances in each pixel using a process called double interpolation. In the proposed framework, the final pixel value would be decided by choosing among different candidates generated by all the input algorithms. Those pixels with highest local consistency would be chosen as the final results. Therefore, the PSNRs of the results of proposed approach may not always be the highest among all conditions, but the color artifact is usually much less than the results of other algorithms. Also, one of the input algorithms in the proposed framework is good at obtaining excellent PSNRs and the other excels at getting very low zipper effect, but the proposed algorithm can outperform them in average both in PSNRs and in zipper effect. The only parameters in this proposed framework are the size and weightings of the decision window, which do not play an important role on the output PSNRs and zipper effect. The larger window provides the lower zipper effect but lower PSNRs and vice versa. The experimental results show that the proposed approach outperforms the other algorithms in both average objective quality assessment and subjective visual quality.

## REFERENCES


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

[2] W. Lu and Y.-P. Tan, "Color filter array demosaicing: new method and performance measures," *IEEE Trans. Image Processing*, vol. 12, no. 10, pp. 1194–1210, Oct. 2003.

[3] B. K. Gunturk, J. Glotzbach, Y. Altunbasak, R. W. Schafer, and R. M. Mersereau, "Demosaicking: color filter array interpolation," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp.44-54, Jan. 2005.

[4] S.-C. Pei and I.-K. Tam, "Effective color interpolation in CCD color filter arrays using signal correlation," *IEEE Trans. on Circuits Sys. Video Tech.*, vol. 13, no. 6, pp.503- 513, Jun. 2003.

[5] X. Li, "Demosaicing by successive approximation," *IEEE Trans. Image Processing*, vol. 14, no. 3, p. 370V379, Mar. 2005.

[6] T.-H. Chen and S.-Y Chien, "Cost effective color filter array demosaicking with chrominance variance weighted interpolation," *IEEE International Symposium on Circuits and Systems*, pp.1277-1280, May 2007.

[7] Chi-Yi Tsai and Kai-Tai Song, "Heterogeneity-projection hard-decision color interpolation using spectral-spatial correlation," *IEEE Trans. Image Processing*, vol.16, no.1, pp.78-91, Jan. 2007.

[8] Kuo-Liang Chung, Wei-Jen Yang, Wen-Ming Yan, and Chung-Chou Wang, "Demosaicing of color filter array captured images using gradient edge detection masks and adaptive heterogeneity-projection," *IEEE Trans. Image Processing*, vol.17, no.12, pp.2356-2367, Dec. 2008.

[9] M.-C. Chuang, Y.-N. Liu, T.-H. Chen, and S.-Y. Chien, "Color filter array demosaicking using joint bilateral filter," in *Proc. IEEE ICME*, pp. 125–128, Jul. 2009.

[10] B. Leung, G. Jeon, and E. Dubois, "Least-squares luma–chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. Image Processing*, vol.20, no.7, pp.1885-1894, Jul. 2011.

[11] A. Hore and D. Ziou, "An edge-sensing generic demosaicing algorithm with application to image resampling," *IEEE Trans. Image Processing*, vol.20, no.11, pp.3136-3150, Nov. 2011.

[12] A. Heinrich, G. de Haan, and C. N. Cordes, "A novel performance measure for picture rate conversion methods," in *Digest of Technical Papers of International Conference on Consumer Electronics (ICCE 2008)*, pp.1-2, Jan. 2008.

[13] Y.-N. Liu, T.-C. Lin, and S.-Y. Chien, "A no reference quality evaluation method for CFA demosaicking," in *Digest of Technical Papers of International Conference on Consumer Electronics (ICCE 2010)*, pp. 365–366, Jan. 2010.

[14] L. Zhang, X. Wu, A. Buades, and X. Li, "Color Demosaicking by Local Directional Interpolation and Non-local Adaptive Thresholding," *Journal of Electronic Imaging* 20(2), 023016, Apr.-Jun. 2011.