IMAGE DEMOSAICKING BASED ON LOGISTIC EDGE-SENSING USING COLOR FILTER ARRAYS

Dr. E. Sree Devi[1] & O. Sugel Anand[2]

[1]Professor of ECE, Rohini College of Engineering & Technology
[2]Research Scholar, Anna University, Chennai

---***---

Abstract - Digital cameras that use Color Filter Arrays (CFA) entail a demosaicking procedure to form full RGB images. To digital camera industry, demosaicking speed is as important as demosaicking accuracy, because camera users have been accustomed to viewing captured photos instantly. Moreover, the cost associated with demosaicking should not go beyond the cost saved by using CFA. For this purpose, image demosaicking based on Logistic Edge-Sensing Demosaicking. Digital cameras that use Color Filter Arrays (CFA) entail a demosaicking procedure to form full RGB images. Therefore propose a very low cost edge sensing scheme, which guides demosaicking by a logistic functional of the difference between directional variations. On test images of currently popular resolution, the quality of our algorithm is comparable to top performers, yet our speed is tens of times faster. Experimental results show that the proposed method outperforms several existing and state-of-the art methods in terms of both subjective and objective evaluations.

Keywords - Demosaicking, Color Filter Array (CFA), Bayer Pattern, Logistic Function

1. INTRODUCTION

Digital images are comprised of data samples arranged in a two dimesional grid. These data samples are usually referred to as picture elements or pixels. The number of pixels in an image determines its resolution. The higher number of pixels an image has, the more information it could contain and the better it could represent the original data. In other words, all other things being equal, a high resolution image has better quality than a low resolution one. Changing the resolution of an image is called image resampling. One may need to resample an image for a variety of reasons. For instance, if a display device has lower resolution than an image to be displayed, then the image needs to be down sampled so that it could fit to the display screen. Similarly, if an image takes up too much data storage space or takes too long to transmit, a possible solution (other than applying compression) is to down sample the image.

While most interpolation algorithms interpolate smooth regions successfully, they tend to fail in edge packed regions. Common interpolation failures are blurriness and staircase effect which refers to edge jagginess. In order to avoid such artifacts, some interpolation algorithms try to detect edge presence and orientation, and adapt the interpolation coefficients accordingly. However, edge detection can be error prone and costly, which results in degraded interpolation performance. propose an edge preserving interpolation method that does not require explicit edge detection. The proposed method studies the relationship between low and high resolution pixels and it applies the same interpolation formula to all input pixels.

Demosaicing or Color Filter Array (CFA) interpolation is a special image interpolation problem. Here, the image size is fixed but only a subset of the color information is available at each pixel location. The mising information at every pixel need to be estimated to obtain the complete color image. While spatial correlation is the only estimation basis for regular image interpolation, spectral correlation between the color channels also comes into play for the demosaicing problem.

Demosaicing algorithms need to exploit both of them to avoid false color artifacts that are closely associated with the demosaicing process. Simple spatially invariant demosaicing methods work in smooth regions with subtle color changes, but they tend to fail around structures with saturated colors.

Adaptive methods that take advantage of local directional information have been introduced to improve the interpolation quality. The work on the demosaicing area have resulted in several algorithms and a new family of CFA patterns.

Linear spatially invariant interpolation techniques such as bilinear and bicubic interpolation have low computational complexity. However, they often fail to protect the integrity of edge structures and introduce blurring. Some adaptive techniques have been proposed to overcome such shortcomings. Transform based algorithms try to extract high frequency information from the image and use it to improve interpolation quality. Although wavelength transform is the most common method of choice, algorithms based on other transforms, such as fourier and discrete cosine, were proposed too. The main drawback of the transform based algorithms is their high computational cost. On top of the transform and inverse transform calculation costs, the iterative nature of these algorithms make them computationally expensive compared to linear interpolation methods.

Another approach to the interpolation problem is to use image statistics for training interpolation filters. The idea is to classify pixels using some kind of feature extraction and to train suitable filters for each pixel class. Resolution Synthesis is an early example of classification based algorithms. Proposed a classification based polynamial interpolation and applied their ideas to temporal interpolation for video sequences. Another method described in uses neural networks to train content-adaptive filters. Edge adaptive interpolation is yet another approach to the interpolation problem.

It became a center of focus because the importance of edge preservation for improved interpolation quality has been recognized early. The New Edge Directed Interpolation (NEDI) proposed is a spatially adaptive interpolation technique that uses local covariance information to preserve the edge structure. Its basic assumption is that there is a significant correlation between low resolution and high resolution local covariances. Once the local covariance for the low resolution image is estimated, it can be used to adapt the interpolation coefficients for that neighborhood. NEDI algorithm is powerful at maintaining defined edges. It does not introduce any sign of staircase effect in most cases. However, the algorithm tends to perform poorly in textured areas especially where closely packed edge structures are present. It also introduces artifacts for perfectly horizontal or vertical edges.

2. METHODOLOGIES

Propose a high-quality fast edge-sensing image demosaicking scheme that adopts the HA work-flow. Particularly, recover the green channel first, and then the green-red and green-blue color difference planes. For adaptive edge-sensing, replace HA's green channel selective directional interpolation by blending the directional estimation, using a logistic functional of the difference between directional variations. The proposed demosaicking process is highly parallelizable: although the red and blue channels have to be estimated subsequently to the green channel estimation, the restoration in each step at a pixel is independent of the restoration of other pixels. This feature means that method is very suitable for Graphics Processing Units (GPU) and Field Programmable Gate Array (FPGA) implementation, achieving instant image visualization175 in high resolution digital cameras.

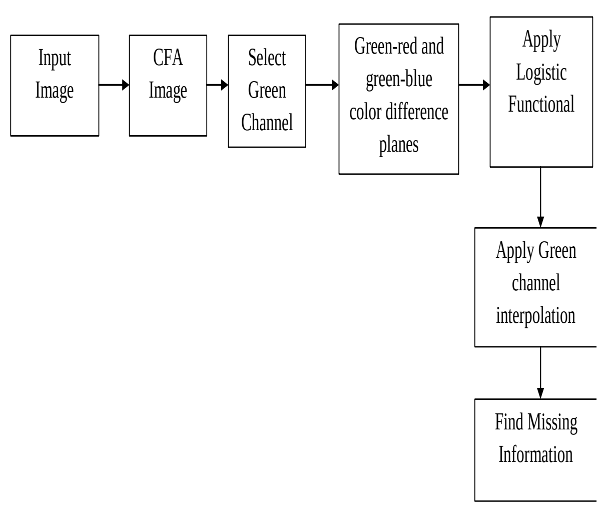

Fig.1.1 Architecture Diagram

A. Input Image

The input image is get from kodak database. The Kodak dataset contains 24 images. The uiget file function is used to select the input image.

B. CFA Image

A color filter array (CFA), is a mosaic of tiny color filters placed over the pixel sensors of an image sensor to capture color information. Color filters are needed because the typical photo sensors detect light intensity with little or no wavelength specificity. The color filters filter the light by wavelength range, such that the separate filtered intensities include information about the color of light. The raw image data captured by the image sensor is then converted to a full-color image.

C. Select Green channel

The information of the green channel is half of the entire sensor and two times of the red or blue channel. During demosaicking process, the rest three-fourths of the full image in the red or blue channel is estimated, while for the green channel only half of the full image is estimated. So the estimation error of the green channel is less than that of the other two.

The green channel demosaicking process of the HA algorithm,

$$\hat{g}(i,j) = \omega_h \left(\bar{g}h - \partial_h^2 M(i,j) + (1-\omega_h)\left(\bar{g}v - \partial_v^2 M(i,j)\right)\right) \quad (1)$$

Where

$$\omega_h = \begin{cases} 0 \ \ if v_h < v_v \\ 1 \ \ if \ v_h > v_v \\ \frac{1}{2} \ if \ v_h = v_v \end{cases} \quad (2)$$

In practice, even in very flat region, v_h and v_v are rarely equal because of noise. A more practical solution is to relax the strict equality requirement $v_h = v_v$ to the approximate equality $v_h \approx v_v$, which can be expressed by the inequality $|v_h - v_v| \leq T$, where T is the allowed noise level. That is,

$$\omega_h = \begin{cases} 0 \ \ if v_h - v_v > T \\ 1 \ \ if \ v_h - v_v < -T \\ \frac{1}{2} \ if \ v_h - v_v \leq T \end{cases} \quad (3)$$

The value of T has to be carefully defined for each image, as a small bias in T may lead to an opposite interpolation decision. Desirably, $\omega_h$should be a continuous function, which smoothly blends the estimation from both directions, thus a small bias does not cause the demosaicking result to vary abruptly. In particular, rather than using the step function defined, seek for a smooth function ω_h that meets the criteria:

1) $\omega_h \to 0$, when $T < v_h - v_v \to \infty$
2) $\omega_h \to 1$, when $-T > v_h - v_v \to -\infty$
3) $\omega_h \approx \frac{1}{2}$, $if \ \ |v_h - v_v| \leq T$

That is, if there is a function f, such that $\omega_h = f(v_h - v_v)$ then $1 - \omega_h = f(v_v - v_h)$ should hold. In other words,

$$f(v_h - v_v) + f(v_v - v_h) = 1 \quad (4)$$

D. Green-red and green-blue color difference planes

Green components at the red and blue sampling positions are interpolated using the proposed algorithm. The blue components can be calculated based on the changing ratio of Green components in the same position. The red components can be calculated based on the changing ratio of Green components in the same position.

Transform r(R^c) and b(B^c) estimation to $(\hat{g} - r)(R^c)$and $(\hat{g} - b)(B^c)$interpolation. treat the two channels in the same fashion, hence this section only articulates the red channel demosaicking. Its blue channel counterpart can be derived by simply exchanging the positions of red and blue in the algorithm. To respect edges and textures, apply edge-sensing strategy also to the red channel. This is cannot be done by a straightforward extension from the green to the red channel. Due to Bayer CFA color sensors arrangement, in the green channel, at a pixel $(i,j) \in G^c$, its horizontal and vertical neighbours all have original green values available. In contrast, if $(i,j) \in R^c$, at most two of its horizontal and vertical neighbours have green-red difference values available.

E. Apply Logistic Functional

Edge-sensing Demosaicking by Logistic Functional of Difference between Directional Variations. First step is to apply Green Channel Demosaicking process. Second step is to apply Red and Blue Channels Demosaicking. To respect edges and textures, apply edge-sensing strategy also to the red channel. This cannot be done by a straightforward extension from the green to the red channel. Due to Bayer CFA color sensors arrangement, in the green channel, at a pixel (i,j) ∈G c, its horizontal and vertical neighbours all have original green values available. In contrast, if (i,j) ∈R c, at most two of its horizontal and vertical neighbours have green-red difference values available.

F. Apply Green channel Interpolation

In the first step of interpolation, the green color channel is interpolated firstly because the green samples are twice as many as those in the red color channel and blue color channel. There are two cases. Case 1 is when the green pixel is at the location of (i,j) , which is can get the green channel directly, which means that A(i,j) is G(i,j). Case 2 is when the nearest neighbors to (i,j) are at the four corners A1(i,j) denotes a red (blue) pixel and A2(i,j) denotes a green (red) pixel relatively, or vice versa.

3. RESULT AND DISCUSSION

Evaluate the proposed algorithm, which we name Logistic Edge-Sensing Demosaicking (LED). To examine the pure effectiveness of steering demosaicking by logistic functional of the difference between directional variation, we do not enhance the image restoration quality by any post-processing or refinement technique.

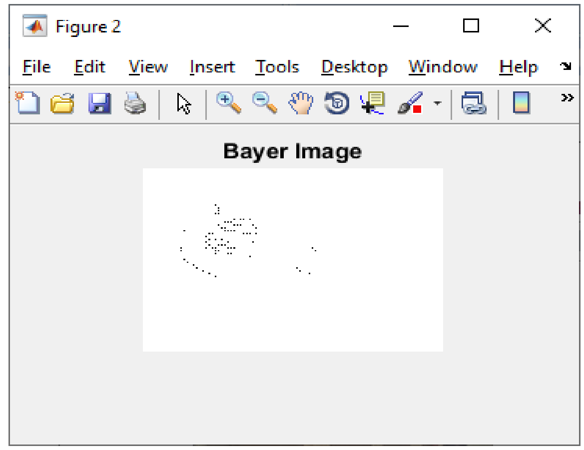

Fig 1.2 Bayer layer

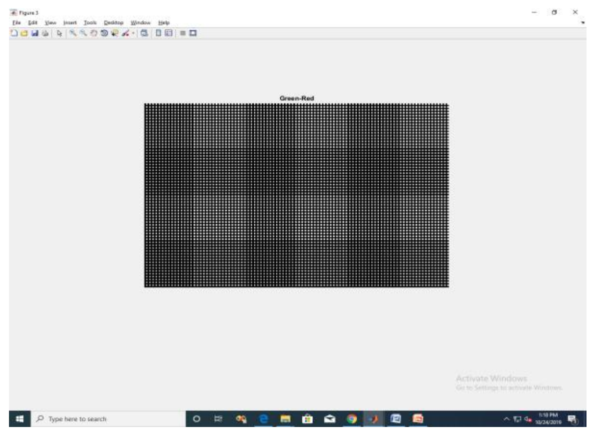

Fig 1.3 Green-Red Difference Plane

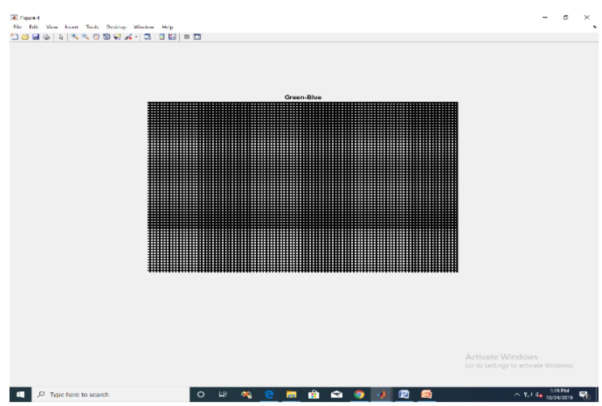

Fig 1.4 Green-Blue Difference Plane

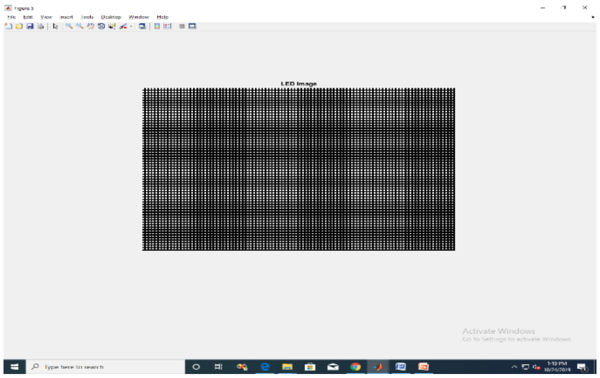

Fig 1.5 LED Image

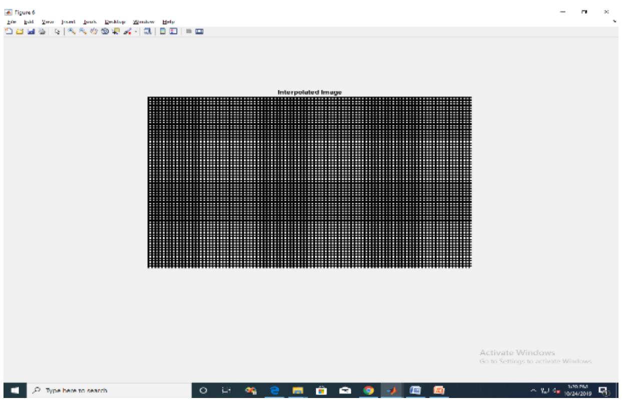

Fig 1.6 Interpolated Image

4. CONCLUSION

Proposed a very low cost edge sensing strategy, termed as LED, for color image demosaicking. It guides the green channel interpolation and color difference plane570 interpolation by logistic functional of the difference between directional variation. Among 28 demosaicking methods tested by code running, method is one of the fastest. Without using any refinement or post-processing technique, LED achieves the accuracy higher than many recently proposed methods on low resolution images, and comparable to top performers on images of currently popular resolution. Extensive experiments suggest that, accurate non-local edge detection for demosaicking is generally difficult and time consuming. Instead, leveraging the originally captured values of the nearest neighbours is much more efficient.

REFERENCES

1. Buades A., Coll B., Morel J.-M., and Sbert C., (2009) “Self-similarity driven color demosaicking,” IEEE Transactions on Image Processing, vol. 18, no. 6, pp. 1192–1202.
2. Chung K. H and Chan Y.-H., (2006) “Color demosaicing using variance of color differences,” IEEE Transactions on Image Processing, vol. 15, no. 10, pp
3. Jr J. F. H.. and Adams J. E., (1997) “Adaptive color plane interpolation in single sensor color electronic camera,” uS Patent 5,629,734.
4. Kiku D., Monno, Y. Tanaka M. and Okutomi M., (2013) “Residualinterpolation for color image demosaicking,” in Proc. IEEE International Conference on Image Processing, pp. 2304–2308.
5. Malvar H. S., He L. and Cutler R., (2004) “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. iii–485–8.
6. Monno Y., Kiku D., Tanaka M. and Okutomi M., (2015) “Adaptive residual625 interpolationforcolorimagedemosaicking,”inPro c.IEEEInternational Conference on Image Processing, pp. 3861–3865.
7. Pekkucuksen I. and Altunbasak Y., (2010) “Gradient based threshold free color filter array interpolation,” in Image Processing, IEEE International Conference on, pp. 137–140.
8. Pekkucuksen I. and Altunbasak Y., (2013) “Multiscale gradients-based color filter array interpolation,” IEEE Transactions on Image Processing, vol. 22, no. 1, pp. 157–165.
9. Shao L. and Rehman A. U., (2014) “Image demosaicing using content and colour-correlation analysis,” Signal Processing, vol. 103, pp. 84–91,600.
10. Szeliski R., (2010) “Computer vision: algorithms and applications”. Springer Science & Business Media.
11. Tajima S., Funatsu R. and Nishida Y., (2014) “Chromatic interpolation based on anisotropy-scale-mixture statistics,” Signal Processing, vol. 97, pp. 262–268.
12. Wu J., Anisetti M., Wu W., Damiani E., and Jeon G., (2016) “Bayer demosaicking with polynomial

interpolation," IEEE Transactions on Image Processing, vol. 25, no. 11, pp. 5369–5382.

13. Wu J., Timofte R. and Gool L. V, (2016) "Demosaicing based on directional difference regression and efficient regression priors," IEEE Transactions on Image Processing, vol. 25, no. 8, pp. 3862–3874.
14. Zhang C., Li Y., Wang J. and Hao P., (2016) "Universal demosaicking of color filter arrays," IEEE Transactions on Image Processing, vol. 25, no. 11, pp. 5173–5186.
15. Zhang L., Wu X., Buades A. and Li X., (2011) "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," Journal of Electronic imaging, vol. 20, no. 2, p. 023016..