Channel-Wise Predictive Filter Flow for Demosaicking

Taishi Iriyama\textsuperscript{1}, Masatoshi Sato\textsuperscript{1}, Hisashi Aomori\textsuperscript{2} and Tsuyoshi Otake\textsuperscript{1}

\textsuperscript{1}Graduate School of Engineering, Tamagawa University
6-1-1 Tamagawagakuen, Machida, Tokyo 194-8610, Japan
\textsuperscript{2}Department of Electrical and Electronic Engineering, Chukyo University
101-2 Yagoto-Honnachi, Showa-ku, Nagoya 466-8666, Japan
E-mail: irymt3sc@engs.tamagawa.ac.jp, otake@ieee.org

Abstract

Demosaicking is an image reconstruction process to recover a full color image from color filter array (CFA) mosaic data. Recently, deep convolutional neural network (CNN)-based demosaicking methods have been explored and have achieved state-of-the-art accuracy. In the deep-CNN-based demosaicking, output pixels are affected by a large spatial region; however, the information involved in demosaicking often only exists locally. In this paper, we propose a channel-wise predictive filter flow (PFF) for demosaicking. Since the PFF is a model that predicts a space-variant linear filter that is transformed to the target image by linearly combining it with the input image, target pixels are reconstructed only from local information. To incorporate the PFF into demosaicking, the proposed network synthesizes the filter flow corresponding to each channel by different networks that are learned independently. Experimental results demonstrate that the proposed method provides better or competitive results compared with several state-of-the-art deep-CNN-based demosaicking algorithms.

1. Introduction

Digital cameras acquire an image by converting light into electrical signals using an image sensor. Digital cameras are commonly equipped with a single image sensor with a color filter array (CFA) to acquire mosaic data with only one-color information at each pixel. The Bayer CFA, which consists of RGB color filters, is the most popular CFA pattern. To recover a full-color image from mosaic data, the missing two-color information must be interpolated \cite{1}. This process is referred to as demosaicking or CFA interpolation. Estimating each color channel separately using spatially invariant standard interpolation techniques results in high-frequency information being lost in the reconstructed image, leading to artifacts such as false colors and the zipper effect. To reduce these artifacts, many demosaicking algorithms have been proposed.

Recently, deep convolutional neural network (CNN) methods have provided superior performance for various image processing tasks such as super-resolution, demosaicking, classification, deblurring and denoising. In deep-CNN-based image processing, global features can be utilized by a wide receptive field, so that the outputs are affected by a large spatial region. However, in image reconstruction tasks such as super-resolution and demosaicking, the information involved in image reconstruction often only exists locally. In this paper, we propose a channel-wise predictive filter flow (PFF) for demosaicking. The PFF predicts weights of a space-variant linear filter to obtain target pixels by linearly combining them with the input image, so that the target pixels are reconstructed only from local information. To incorporate the PFF into demosaicking, we propose a novel framework that predicts the different filter flows corresponding to each output channel.

The contributions of this paper are summarized as follows:
(1) A channel-wise PFF for demosaicking is proposed and a novel framework for incorporating the PFF into the demosaicking is shown. (2) Experimental results on the Kodak and McMaster datasets demonstrate that the proposed method provides better or competitive results compared with state-of-the-art deep-CNN-based demosaicking algorithms.

2. Predictive Filter Flow (PFF)

The PFF was proposed as a model that predicts the optimal filter flow using a CNN \cite{2}. The filter flow is a space-variant linear filter used to reconstruct the target image by linearly combining it with the input image \cite{3}. In the PFF, the filter flow is predicted through the CNN, and output pixels depend only on the local neighborhood centered on the input image at the same coordinate as the target pixel. The filter flow $T$ is obtained by optimizing the parameter $\theta$ of the network $F(\cdot)$ from the given input image $I_1$ and target image $I_2$.

\begin{equation}
I_2 \approx TI_1, T \equiv F_\theta(I_1)
\end{equation}
The network $F()$ is trained by minimizing the $l_1$-norm loss between the predicted image and the target image,

$$loss = \sum_{i=0}^{N} |F(I_1) \cdot I_1 - I_2|_1$$

where $N$ indicates the number of training observations.

3. Channel-Wise PFF Network

To incorporate the PFF into the demosaicking, we design a novel demosaicking framework with a channel-wise PFF illustrated in Fig. 1. The proposed demosaicking framework consists of two parts, feature extraction and channel-wise filter flow synthesis. In the first part, the features are efficiently extracted by an ensemble of deep and shallow CNNs. In the second part, the channel-wise filter flows are obtained by using different networks corresponding to each output channel.

3.1 Feature extraction

As shown in Fig. 1, the feature extraction consists of a two-stream network of deep and shallow CNNs. The deep CNN extracts the global and complex features, and the shallow CNN extracts the local features and retains the spatial information of the Bayer pattern CFA image. The proposed method adopts the residual-in-residual (RIR) structure proposed in [4] as the deep CNN. The RIR (Fig. 2 (a)) consists of a long skip connection and several residual groups (RGs), and each RG consists of a short skip connection and several residual blocks (RB) (Fig. 2 (b)). The RIR can train the very deep-CNN using the residual structure with the long and short skip connections. The shallow CNN (Fig. 2 (c)) consists of $1 \times 1$ and $3 \times 3$ convolution layers with a ReLU layer.

3.2 Channel-wise filter flow synthesis

From the features obtained by concatenating the deep and shallow CNNs, filter flows for each channel are synthesized by a 3-layer network consisting of $3 \times 3$ convolutional, batch normalization and ReLU layers (Fig. 3). These convolution layers output a 3-dimensional array with the channel dimension of the filter size as the linear filter weight corresponding to each spatial dimension. In order for each filter to maintain the output intensity, the softmax function is applied to the last layer instead of the ReLU layer. Filter flow synthesis networks are prepared for each channel, and these networks do not share weights and are learned independently. The prediction image is obtained by multiplying the synthesized channel-wise filter flows and the input mosaicked image reshaped by using the “im2col” operation, and summing the elements in the channel direction.
4. Experiments

For training, we use the Waterloo Exploration Database (WED) dataset [5] consisting of 4744 color images. The training dataset is randomly cropped to 64 × 64-size patches and downsampled to the Bayer pattern CFA. The hyperparameters in the training of our proposed network are set as follows: the filter size is 25 × 25, the number of RBs in the RIR is 10, the number of RBs in an RG is 20, the mini-batch size is 10, the initial learning rate of the Adam optimizer [6] is $10^{-4}$ and the number of epochs is 500. For testing, we use the Kodak dataset [7] consisting of 24 color images (768 × 512) and the McMaster dataset [8] consisting of 18 color images (500 × 500). Our proposed network was implemented in PyTorch with an NVIDIA 1080 GPU.

4.1 Quantitative result

As a quantitative comparison, Table 1 shows the demosaicking results of the proposed method and two state-of-the-art CNN-based demosaicking methods, 3-stage CNN structure (3-Stage) [9] and lightweight deep residual learning (LDRL) [10], for the color peak signal-to-noise ratio (CPSNR) and structural similarity (SSIM) measures. As shown in Table 1, the proposed method shows a higher SSIM on the Kodak dataset and higher CPSNR and SSIM on the McMaster dataset than the other demosaicking methods. The average execution times to reconstruct the Kodak and McMaster images using the proposed method are approximately 3.587 s and 2.398 s, respectively.

### Table 1: Quantitative comparison of CPSNR and SSIM

| No. | 3-Stage | LDRL | Proposed | No. | 3-Stage | LDRL | Proposed |
|-----|---------|------|----------|-----|---------|------|----------|
|     | CPSNR   | SSIM |          |     | CPSNR   | SSIM |          |
| 01  | 41.81   | 0.9952 | 42.41 | 0.9940 | 41.15 | 0.9951 | 13 | 37.50 | 0.9924 | 38.04 | 0.9912 | 37.43 | 0.9923 |
| 02  | 42.52   | 0.9987 | 42.11 | 0.9822 | 41.92 | 0.9924 | 14 | 40.15 | 0.9922 | 40.96 | 0.9923 | 40.77 | 0.9949 |
| 03  | 45.23   | 0.9954 | 45.51 | 0.9919 | 45.24 | 0.9965 | 15 | 42.49 | 0.9917 | 42.03 | 0.9861 | 42.27 | 0.9941 |
| 04  | 43.61   | 0.9974 | 43.74 | 0.9895 | 43.36 | 0.9946 | 16 | 45.58 | 0.9926 | 45.82 | 0.9928 | 45.08 | 0.9963 |
| 05  | 40.88   | 0.9951 | 41.24 | 0.9945 | 41.00 | 0.9960 | 17 | 42.86 | 0.9824 | 43.22 | 0.9910 | 43.22 | 0.9958 |
| 06  | 42.50   | 0.9956 | 42.89 | 0.9925 | 41.45 | 0.9951 | 18 | 39.23 | 0.9795 | 39.49 | 0.9869 | 39.08 | 0.9921 |
| 07  | 45.09   | 0.9965 | 45.17 | 0.9931 | 44.99 | 0.9971 | 19 | 42.83 | 0.9834 | 43.07 | 0.9889 | 41.11 | 0.9937 |
| 08  | 39.14   | 0.9893 | 39.18 | 0.9915 | 39.48 | 0.9947 | 20 | 43.32 | 0.9563 | 43.16 | 0.9821 | 42.72 | 0.9931 |
| 09  | 44.47   | 0.9771 | 44.73 | 0.9869 | 44.69 | 0.9955 | 21 | 41.55 | 0.9917 | 41.90 | 0.9859 | 41.33 | 0.9939 |
| 10  | 44.27   | 0.9868 | 44.36 | 0.9885 | 44.24 | 0.9955 | 22 | 40.42 | 0.9856 | 40.94 | 0.9851 | 40.52 | 0.9917 |
| 11  | 42.39   | 0.9838 | 42.76 | 0.9917 | 41.82 | 0.9946 | 23 | 45.00 | 0.9955 | 45.36 | 0.9889 | 44.92 | 0.9960 |
| 12  | 45.38   | 0.9968 | 45.56 | 0.9907 | 45.00 | 0.9958 | 24 | 37.28 | 0.9867 | 37.73 | 0.9909 | 37.75 | 0.9941 |

Kodak Ave. 42.31 0.9891 42.56 0.9895 42.11 0.9946
Test Time Ave. 12.701[ms] 2.086[ms] 3.587[ms]

| No. | 3-Stage | LDRL | Proposed | No. | 3-Stage | LDRL | Proposed |
|-----|---------|------|----------|-----|---------|------|----------|
|     | CPSNR   | SSIM |          |     | CPSNR   | SSIM |          |
| 01  | 31.50   | 0.9788 | 31.52 | 0.9451 | 32.16 | 0.9681 | 10 | 41.18 | 0.9868 | 41.14 | 0.9798 | 41.69 | 0.9919 |
| 02  | 36.28   | 0.9678 | 36.51 | 0.9576 | 36.95 | 0.9904 | 11 | 41.99 | 0.9912 | 41.82 | 0.9791 | 42.32 | 0.9922 |
| 03  | 36.56   | 0.9676 | 36.64 | 0.9805 | 37.27 | 0.9907 | 12 | 41.78 | 0.9982 | 41.69 | 0.9726 | 41.76 | 0.9905 |
| 04  | 40.96   | 0.9635 | 40.60 | 0.9925 | 41.70 | 0.9970 | 13 | 42.51 | 0.9978 | 42.49 | 0.9630 | 42.59 | 0.9886 |
| 05  | 37.15   | 0.9803 | 36.99 | 0.9712 | 37.85 | 0.9869 | 14 | 40.43 | 0.9852 | 40.52 | 0.9673 | 40.86 | 0.9875 |
| 06  | 41.21   | 0.9764 | 40.87 | 0.9785 | 41.58 | 0.9913 | 15 | 40.74 | 0.9875 | 40.60 | 0.9667 | 41.02 | 0.9878 |
| 07  | 41.52   | 0.9633 | 41.77 | 0.9853 | 41.91 | 0.9933 | 16 | 37.35 | 0.9933 | 37.01 | 0.9742 | 38.01 | 0.9982 |
| 08  | 41.30   | 0.9199 | 41.39 | 0.9822 | 41.89 | 0.9934 | 17 | 36.90 | 0.9915 | 36.65 | 0.9747 | 37.65 | 0.9881 |
| 09  | 40.54   | 0.9935 | 40.21 | 0.9729 | 40.73 | 0.9906 | 18 | 38.23 | 0.9787 | 38.01 | 0.9723 | 38.40 | 0.9969 |

McMaster Ave. 39.34 0.9790 39.25 0.9731 39.80 0.9885
Test Time Ave. 7.491[ms] 1.774[ms] 2.398[ms]
5. Conclusions

A channel-wise PFF for demosaicking is proposed. The proposed method synthesizes a channel-wise filter flow using different networks that are learned independently to incorporate the PFF into demosaicking. In accordance with experimental results, the proposed method provides better or competitive results on benchmark datasets compared with other deep-CNN-based demosaicking techniques. In particular, the quantitative result demonstrates a higher SSIM because the PFF reconstructs target pixels using only the local neighborhood pixels.

References

[1] R. Ramanath and W. E. Snyder et al.: Demosaicking methods for Bayer color arrays, Journal of Electronic Imaging, Vol.11, No.3, pp.306–316, 2002.
[2] S. Kong and C. Fowlkes: Image reconstruction with predictive filter flow, arXiv preprint arXiv:1811.11482, 2018.
[3] S. M. Seitz and S. Baker: Filter flow, Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.143–150, 2009.
[4] Y. Zhang and K. Li et al.: Image super-resolution using very deep residual channel attention networks, Proc. European Conference on Computer Vision, pp.286–301, 2018.
[5] K. Ma and Z. Duanmu et al.: Waterloo exploration database: New challenges for image quality assessment models, IEEE Transactions on Image Processing, Vol.26, No.2, pp.1004–1016, 2017.
[6] D. Kingma and J. Ba et al.: Adam: A method for stochastic optimization, Proc. International Conference on Learning Representations, pp.1–13, 2015.
[7] Eastman Kodak Company: Low Resolution Kodak Image Dataset. http://r0k.us/graphics/kodak/, 1991.
[8] L. Zhang and X. Wu et al.: Color demosaicking by local directional interpolation and non-local adaptive thresholding, Journal of Electronic Imaging, Vol.20, No.2, pp.023016-1–023016-16, 2011.
[9] K. Cui and Z. Jin: Color image demosaicking using a 3-stage convolutional neural network structure, Proc. IEEE International Conference on Image Processing, pp.2177–2181, 2018.
[10] T. Huang and F. Wu et al.: Lightweight deep residue learning for joint color image demosaicking and denoising, Proc. International Conference on Pattern Recognition, pp.127–132, 2018.