

D³Net: Joint Demosaicking, Deblurring and Deringing

Tomáš Kerepecký

*The Czech Academy of Sciences,
Institute of Information Theory and Automation
Pod Vodárenskou věží 4, Prague, Czechia
Email: <http://www.utia.cas.cz/people/kerepeck>*

Filip Šroubek

*The Czech Academy of Sciences,
Institute of Information Theory and Automation
Pod Vodárenskou věží 4, Prague, Czechia
Email: <http://www.utia.cas.cz/people/sroubek>*

Abstract—Images acquired with standard digital cameras have Bayer patterns and suffer from lens blur. A demosaicking step is implemented in every digital camera, yet blur often remains unattended due to computational cost and instability of deblurring algorithms. Linear methods, which are computationally less demanding, produce ringing artifacts in deblurred images. Complex non-linear deblurring methods avoid artifacts, however their complexity imply offline application after camera demosaicking, which leads to sub-optimal performance. In this work, we propose a joint demosaicking deblurring and deringing network with a light-weight architecture inspired by the alternating direction method of multipliers. The proposed network has a transparent and clear interpretation compared to other black-box data driven approaches. We experimentally validate its superiority over state-of-the-art demosaicking methods with offline deblurring.

Index Terms—Demosaicking, deblurring, deringing, ADMM, CNN

I. INTRODUCTION

Data acquired by modern digital camera sensors are subject to various types of signal degradation, such as lens and sensor blur, aberrations, color filter array (CFA) and noise. To convert the raw data from the imaging sensor into an image suitable for the human visual system, it is necessary to correctly process the acquired data, particularly by applying demosaicking and deblurring procedures. Sequential demosaicking and deblurring provides sub-optimal solutions [1], yet they are still used for their simplicity. Joint demosaicking and deblurring was studied earlier [1]–[4] using traditional model-based optimization approaches. More recent learning-based methods focus only on joint demosaicking and denoising [5]–[8] and disregard blur, which is present in DSLR and mobile phone cameras even if the lens is in focus, see Fig. 2.

An important, yet often neglected, property of restoration algorithms is their ability to run in the camera with limited computational capacity, such as pixel-wise operations and basic filtering. In this regard, a computationally efficient algorithm for deconvolution was proposed in [9]. The algorithm is based on the alternating direction method of multipliers (ADMM) [10] and performs deblurring by iterative Wiener filtering and thresholding (IWFT). Removing blur with the

This work was supported by Czech Science Foundation grant GA20-27939S and by the Praemium Academiae awarded by the Czech Academy of Sciences.

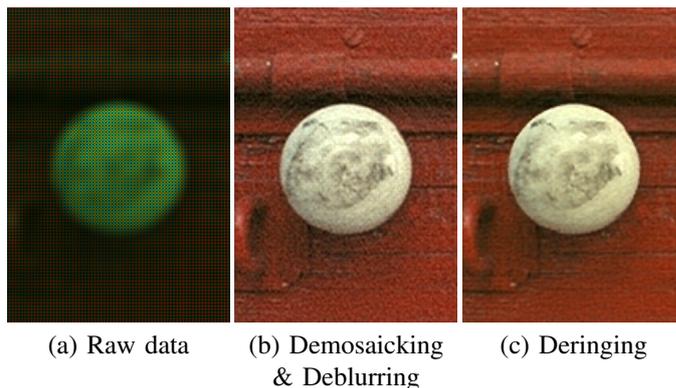


Fig. 1. The proposed convolutional neural network joints three restoration tasks: demosaicking, deblurring and deringing.

Wiener filter (ideal linear filter) produces mediocre results in most of the cases due to ringing artifacts around edges in the image. The IWFT algorithm instead uses two sets of filters, one for the initial restoration (deblurring) and another for the ringing artifact suppression (deringing). These filters are precomputed offline for the given type of degradation, i.e. blur and noise level.

Recent works have revealed that, with the aid of model-based optimization methods, such as Primal-Dual or ADMM, it is possible to design convolutional neural networks (CNN) with clear interpretation [8], [11]. Inspired by these studies, we design a light-weight CNN imitating the IWFT concept, which is directly applicable to raw camera data (Fig. 1). The proposed network – called D³Net – performs joint demosaicking, deblurring and deringing. Network filters have clear interpretation and they become learnable parameters, which is an important advantage over the IWFT algorithm. A relatively small number of training parameters allows us to efficiently train the network by only a single pair of degraded and ground-truth images. We perform quantitative and qualitative evaluation of D³Net and compare it with state-of-the-art demosaicking methods with and without offline deblurring.

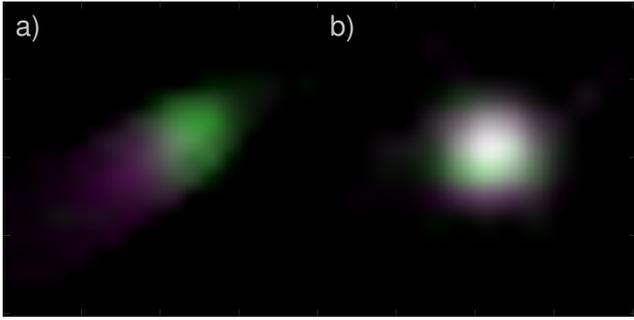


Fig. 2. Intrinsic camera blur (combination of sensor blur and lens aberrations): a) DSLR, b) mobile phone.

II. PROBLEM FORMULATION

To solve the joint demosaicking-deblurring problem, one of the most frequently used approaches in the literature relies on the following linear observation model

$$g = SHu + n, \quad (1)$$

where $g \in \mathbb{R}^p$ is the blurred, noisy raw image and $u \in \mathbb{R}^m$ is the unknown high-resolution sharp image. Both u and g correspond to the vectorized forms of the images. $H(\cdot) \equiv h * \cdot$ denotes a degradation operator (matrix) performing convolution with some known point spread function (PSF) h . For simplicity, we employ a stationary blur model. S represents the down-sampling operator, which models the particular CFA pattern. It corresponds to a binary matrix which excludes the spatial and channel location in the image where color information is missing. We consider additive white Gaussian noise $n \approx \mathcal{N}(0, \sigma^2)$ with zero mean and variance σ^2 .

To solve the ill-posed inverse problem, we adopt the optimization problem with total variation regularization [12]:

$$\hat{u} = \arg \min_u \frac{\gamma}{2} \|SHu - g\|_2^2 + \phi^1(\{D_j u\}), \quad (2)$$

where the norm of the first term is the classical ℓ_2 -norm. $\phi^s(\{D_j u\}) = \sum_i (\sum_j [D_j u]_i^2)^{s/2}$ represents the regularization function. $D_j(\cdot) \equiv d_j * \cdot$ denotes the j -th feature extraction operator implemented as a convolution with the filter d_j . For example, if the set $\{d_j\}$ comprises only vertical and horizontal differences, $\{D_j u\}$ corresponds to the discrete image gradient and $\phi^1(\cdot)$ is the sum of gradient magnitudes. Pixels are indexed as $[u]_i$. Parameter γ is the weight between the data term and regularization.

A popular choice for solving such non-smooth convex problems is ADMM [10]. The method introduces auxiliary variables $v_j \in \mathbb{R}^m$ and equality constraints $v_j = D_j u$, and rewrites (2) as a saddle-point problem for an ‘augmented Lagrangian’:

$$\min_{u, \{v_j\}} \frac{\gamma}{2} \|SHu - g\|_2^2 + \phi^1(\{v_j\}) + \frac{\beta}{2} \phi^2(\{D_j u - v_j - a_j\}), \quad (3)$$

where $a_j \in \mathbb{R}^m$ represents the Lagrange multiplier.

In order to solve joint demosaicking-deblurring minimization problem (3) as well as to deal with ringing artifacts

Algorithm 1 Joint demosaicking, deblurring and deringing

Input: g – blurred image, N – number of iterations,

$\{r_k\}$ – set of restoration filters, $\{w_j\}$ – set of update filters

Output: u – sharp image

- 1: **Initial estimation with restoration filter:**
 - 2: $u_r \leftarrow P(\{r_k * g\})$ [rConv]
 - 3: $i \leftarrow N$, $\{a_j\} \leftarrow 0$, $\beta \leftarrow 10 \max(g)$,
 $u \leftarrow u_r$
 - 4: **repeat**
 - 5: $\tilde{v}_j \leftarrow d_j * u \quad \forall j$ [gConv]
 - 6: **Soft thresholding:**
 - 7: $v_j \leftarrow \text{SoftThr}\left(\tilde{v}_j - a_j, \frac{1}{\beta}\right) \quad \forall j$ [Soft]
 - 8: **Update the Lagrange multiplier:**
 - 9: $a_j \leftarrow a_j + (v_j - \tilde{v}_j) \quad \forall j$ [Add]
 - 10: **Improve the image with update filter:**
 - 11: $u \leftarrow u_r + \sum_j w_j * (v_j + a_j)$ [uConv]
 - 12: $i \leftarrow i - 1$
 - 13: **until** $i = 0$
-

after deconvolution, we imitate IWFT concept and propose computationally efficient algorithm summarized in Alg. 1. See Algorithm 2 in [9] for more details.

ADMM sequentially performs alternating minimization with respect to u and $\{v_j\}$. Minimization over v_j leads to soft thresholding with the threshold $1/\beta$ (line 7). Parameter β is set to 10-times the range of intensity values of the blurred image g . In the case of minimization over u (line 11), the update step can be written as

$$u = P(\{r_k * g\}) + \sum_j w_j * (v_j + a_j), \quad (4)$$

where $\{r_k\}$ and $\{w_j\}$ are the sets of restoration and update filters, respectively. Operator P performs pixel shuffling to assemble the final RGB image. These filters are inputs to the algorithm and they are precomputed offline, similarly as in [9], for the given type of degradation, i.e. blur, CFA pattern and noise level. The Lagrange multiplier a_j is updated by the term $(v_j - d_j * u)$ (line 9).

III. PROPOSED NETWORK ARCHITECTURE

Alg. 1 consists of only filtering and element-wise operations and therefore can be used to design the architecture of the light-weight convolutional neural network. As a result, all convolutional filters in the algorithm become learnable parameters.

The architecture of our proposed network D³Net is shown in Fig. 3. The first filtering step (Alg. 1 - line 2) provides the initial estimator of reconstructed image u , which corresponds to the demosaicking and deblurring tasks. The remaining steps in Alg. 1 (line 5 - 12) are run iteratively and perform the ringing artifact suppression (deringing task). Experimentally we have validated that three iterations provide balanced results between ringing artifact suppression and over-regularization.

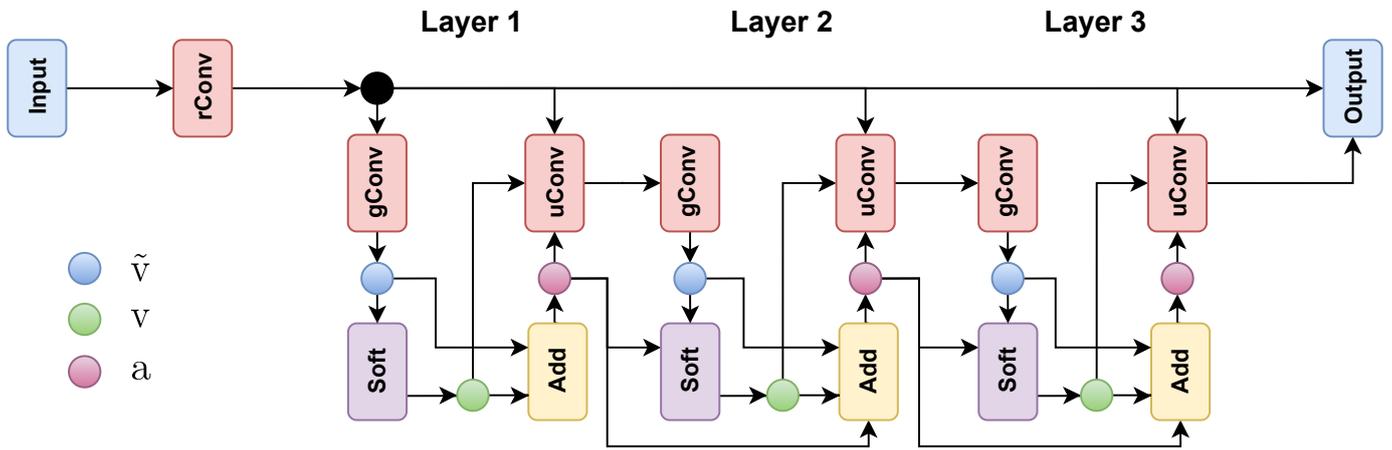


Fig. 3. Proposed network architecture of the D³Net. See Alg. 1 for the interpretation of individual blocks.

The restoration layer (rConv) is therefore followed by three update layers consisting of four operations: gradient filtering (gConv), soft thresholding with the threshold $1/\beta$ (Soft), element-wise operation (Add) and update convolutional layer (uConv).

The input to our network is the degraded blurred image g (Fig. 1a) separated into four channels according to the Bayer CFA pattern. The network can be modified to any other CFA (e.g. X-Trans). The restoration layer (rConv) consists of convolution with 12 four-channel filters $\{r_k\}$ followed by pixel shuffling P which results in the restored RGB image u_r (Fig. 1b) with three times more data than g . An example of one out of 48 restoration filters for demosaicking and deconvolution is in Fig. 4. Filters are initialized using the result of IWFT algorithm (Fig. 4a) and further improved (Fig. 4b) by training with the traditional back-propagation process. The difference between the restoration filter before and after the training is demonstrated in Fig. 4c. Filter size is the hyper parameter of the network and is set, in our example, to 25×25 . The human visual system is more sensitive to high frequencies in the luminance channel, therefore restored RGB image is transformed to a YCbCr color space and all update layers are applied only on luminance channel of the restored image u_r . The complexity of update layers depends on the number of gradient filters. In our implementation, we use 4 filters in the convolutional layers (gConv) which correspond to horizontal, vertical and two diagonal difference operators (Fig. 5a). Consequently, there are four update filters for each update layer (Fig. 6a). Gradient and update filters are also initialized by IWFT. Their modification through the learning procedure is more significant than in the case of restoration filters. The final output of the network (Fig. 1c) is the reconstructed image u , which is the restored luminance channel with chrominance channels transformed back to RGB color space. In total, the proposed network contains 36 convolutions in seven convolutional blocks.

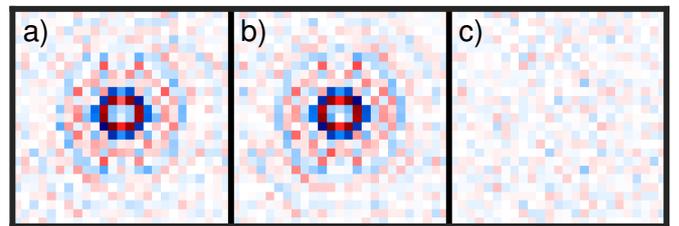


Fig. 4. An example of restoration filter (25×25) for demosaicking and deblurring Bayer data distorted by out-of-focus blur. There are 48 restoration filters in layer rConv. a) initialization from IWFT, b) learned by D³Net, c) difference. Blue-white-red colormap represents numbers from -1 to 1.

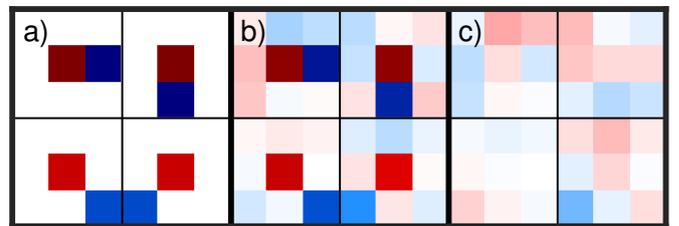


Fig. 5. An example of horizontal, vertical and two diagonal gradient filters (3×3) in layer gConv for deringing images distorted by out-of-focus blur. a) initialization from IWFT, b) learned by D³Net, c) difference. Blue-white-red colormap represents numbers from -1 to 1.

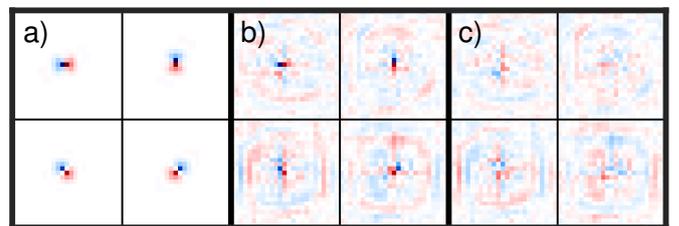


Fig. 6. An example of 4 update filters (25×25) in layer uConv for deringing images distorted by out-of-focus blur. The update filters are related to gradient filters. a) initialization from IWFT, b) learned by D³Net, c) difference. Blue-white-red colormap represents numbers from -1 to 1.

IV. EXPERIMENTS

We use Pytorch as our framework for implementing D³Net. First, we demonstrate the improvement of results from our network over augmented IWFT. In addition, we focus on deringing effect of the proposed network as well as to show the influence of different blurs on final reconstruction. Then we compare our results with sequential demosaicking and deconvolution methods. Finally, we test our network on real images. Throughout the experiments, two objective quality measures were used: Peak Signal to Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM).

A. IWFT vs. D³Net

For training and evaluation of the proposed network, we used publicly available Kodak PhotoCD image dataset. One image from the set was used as a training set and the remaining 23 images formed a validation set. In this experiment, input images were randomly cropped into patches of size 200×200 pixels. We converted the Kodak images into blurred Bayer images by performing an image degradation process (1). Blurred Kodak images were down-sampled with the Bayer pattern GRBG and finally Gaussian noise was added.

Batch size was set to 4. The network was optimized with the mean-squared-error loss. All weights were initialized by filters of the IWFT algorithm, computed similarly as in [9]. Optimization was carried out using the stochastic gradient descent algorithm with learning rate 0.01 and momentum 0.9. Training was super fast with only one epoch, which corresponds to approximately 3.5 minutes on a GeForce RTX 2080 Ti.

We tested both methods, IWFT and D³Net, on out-of-focus blur represented by circular PSF with radius 5 and noise levels 30dB and 40dB. Size of the restoration and update filters were the same and ranged from 5×5 to 35×35 . Size of the gradient filters was 3×3 . Fig. 7 demonstrates the improvement of the results using the learning-based approach. It can be concluded that proposed network gives significantly better PSNR results than IWFT for all filter sizes and noise levels. For the given out-of-focus blur with radius 5, the performance of both methods flattens out for filter sizes of 25×25 and more.

B. Deringing effect

As discussed in Sec. III, update filters change through training more than restoration filters. Therefore we analyzed the performance of update layers, mainly their deringing effect. To train our network, images from Kodak dataset were degraded in the same way as in the Sec. IV-A. This time we used two types of blur: out-of-focus blur with radius 5 and Gaussian blur with variance 3. To form training set with 582930 degraded and ground-truth image pairs, we used 18 images from Kodak dataset and cropped them into patches of size 100×100 . The remaining six Kodak images composed validation set. We considered Gaussian noise 40dB and filter sizes 25×25 . Other parameters remained the same as in the

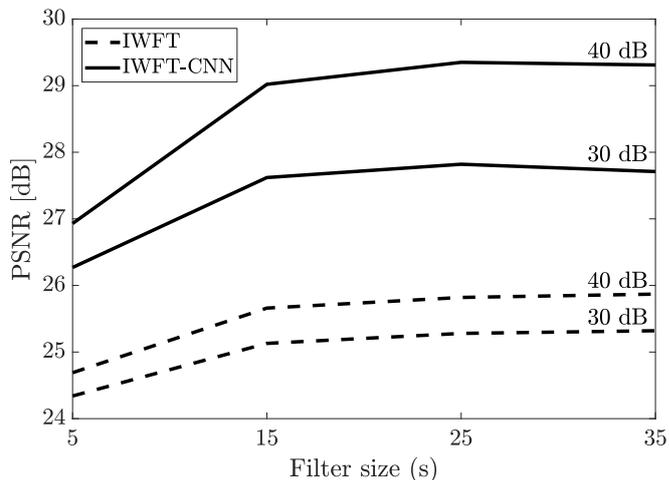


Fig. 7. Average PSNR performance of the proposed D³Net (solid) and IWFT (dashed) [9] with respect to the size (s) of the restoration and update filters ($\{r_k\}, \{w_j\}$). Out-of-focus blur and noise levels with 30dB and 40dB are considered. Proposed network outperforms IWFT for all filter sizes.

previous case. Training of our network lasted approximately 9 hours on a GeForce RTX 2080 Ti.

Tabs. I and II) compare average PSNR and SSIM of the reconstructed images for D³Net, standard demosaicking method [13], Wiener filter and IWFT algorithm in the case of out-of-focus blur and Gaussian blur, respectively.

TABLE I
OUT-OF-FOCUS BLUR: AVERAGE PSNR AND SSIM RESULTS FOR DIFFERENT RECONSTRUCTION METHODS.

| Method | PSNR [dB] | SSIM |
|-------------------------------|--------------|--------------|
| Demosaicked [13] | 24.69 | 0.757 |
| Wiener | 26.10 | 0.874 |
| IWFT | 25.82 | 0.823 |
| D ³ Net (proposed) | 29.94 | 0.926 |

TABLE II
GAUSSIAN BLUR: AVERAGE PSNR AND SSIM RESULTS FOR DIFFERENT RECONSTRUCTION METHODS.

| Method | PSNR [dB] | SSIM |
|-------------------------------|--------------|--------------|
| Demosaicked [13] | 24.53 | 0.752 |
| Wiener | 24.34 | 0.780 |
| IWFT | 25.27 | 0.856 |
| D ³ Net (proposed) | 26.89 | 0.870 |

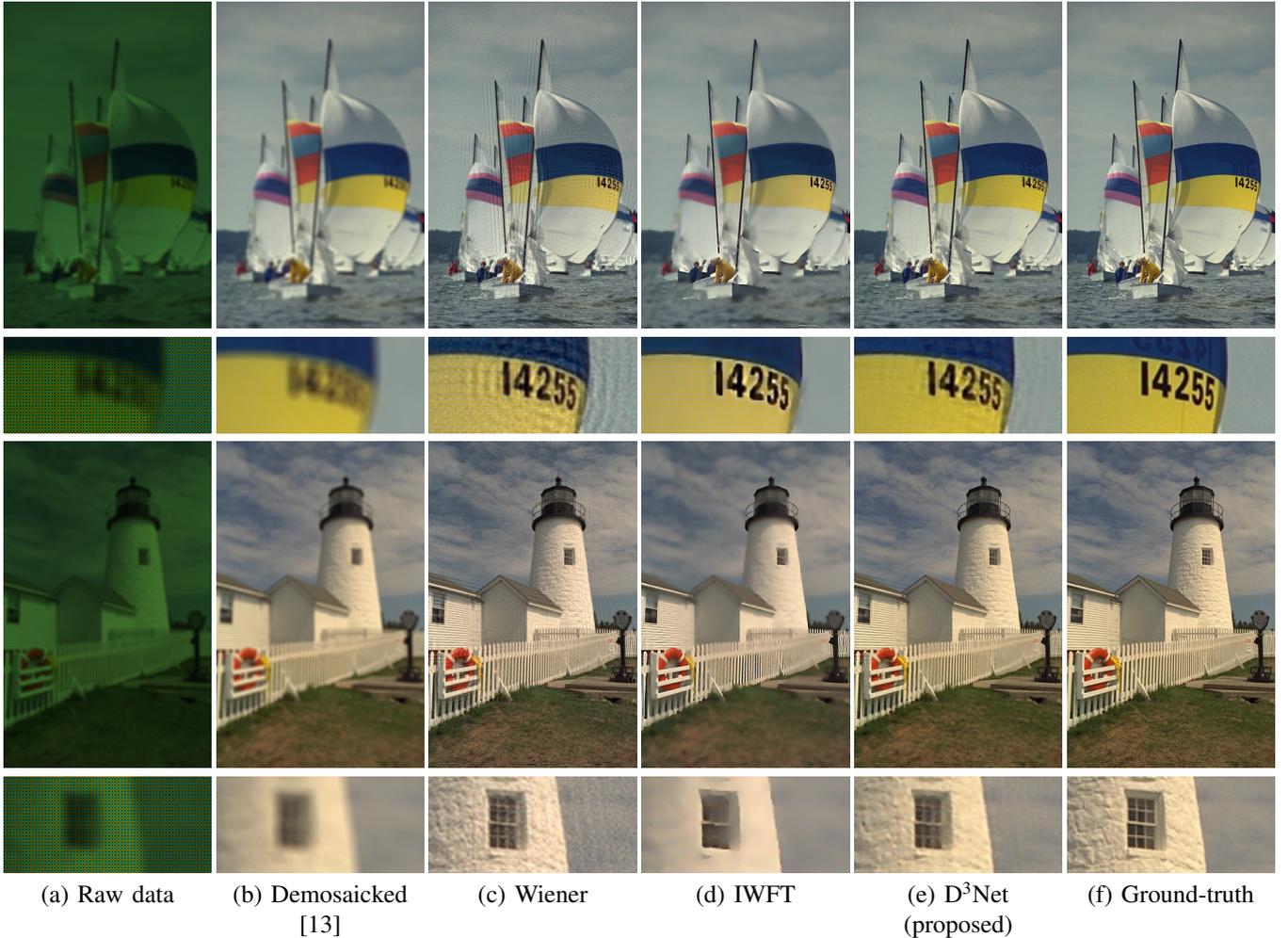


Fig. 8. Visual results from the Kodak dataset. (a) degraded data by out-of-focus blur, CFA Bayer pattern and Gaussian noise with 40dB, (b) applying only demosaicking [13], (c) Wiener filtering, (d) joint demosaicking and deblurring using IWFT method [9], (e) joint demosaicking and deblurring using our proposed network, (f) original sharp image. Our proposed method D³Net retain fine details as opposed to IWFT method that over-smooths highly textured areas while suppresses ringing artifacts when only Wiener filtering is considered.

Wiener filter is equivalent to initial restoration in the IWFT algorithm (i.e. layer rConv in D³Net without learning). It is a popular deconvolution method, however, as a linear filter, the estimated image exhibits ringing artifacts around edges (Fig 8c). Non-linear update steps of IWFT algorithm performed well in suppressing ringing artifacts, hence results were visually better. However, such reconstructed images were over-smoothed (Fig. 8d). This eventually led to lower PSNR and SSIM values for IWFT than for Wiener filter when out-of-focus blur was considered (Tab. I). It was not the case for images blurred by Gaussian PSF, although images remained too smoothed as can be seen in Fig. 9d. Images corrected by D³Net did not suffer from these problems. In Fig. 8e details of the wall are still recognizable as opposed to IWFT. Overall, the proposed network was able to recover more realistic details than the optimization-based IWFT as well as produce images with less visually disturbing artifacts than Wiener-like filters.

C. Joint vs. sequential approach

This experiment presents the comparison of the proposed joint approach with sequential demosaicking and deblurring procedures and evaluates the effect of training the proposed network on more than one image pair. The network trained in Sec. IV-B using 18 Kodak images is denoted D³Net v2 and the network trained on a single pair of degraded and ground-truth image is D³Net v1. We evaluated our networks on the McMaster dataset [14] and compared them with recent demosaicking methods FlexISP [3], DeepJoint [15] and JointADMM [5] followed by robust non-blind deconvolution method (non-blind deconvolution step in [16]). The kernel part of those algorithms, including all parameters, remains the same as their authors provided. Applying offline deblurring is identified by the asterisk symbol *.

An interesting result is that the reconstructed image provided by standard IWFT as well as Wiener filter (Fig. 10e-f) looks visually better and retain relatively fine details as opposed to the other sequential demosaicking and deblurring



Fig. 9. Visual results from the Kodak dataset. (a) degraded data by Gaussian blur, CFA Bayer pattern and Gaussian noise with 40dB, (b) applying only demosaicking [13], (c) Wiener filtering, (d) joint demosaicking and deblurring using IWFT method [9], (e) joint demosaicking and deblurring using our proposed network, (f) original sharp image.

methods, yet they received worse PSNR values (Tab. III).

From Tab. III we observe that D³Net yields substantially better results than all other tested methods. Surprisingly, even network trained on a single image pair (D³Net v1) outperforms sequential demosaicking and deblurring. Our methods leads to better and more visually pleasing results, as it can be seen in Fig. 10l.

TABLE III
OUT-OF-FOCUS BLUR: AVERAGE PSNR AND SSIM RESULTS FOR THE DIFFERENT RECONSTRUCTION METHODS.

| Method | PSNR [dB] | SSIM |
|-----------------------|--------------|--------------|
| JointADMM | 23.06 | 0.742 |
| DeepJoint | 23.40 | 0.751 |
| FlexISP | 23.37 | 0.763 |
| Wiener | 23.57 | 0.826 |
| IWFT | 24.07 | 0.843 |
| JointADMM* | 25.44 | 0.839 |
| DeepJoint* | 25.62 | 0.846 |
| FlexISP* | 26.48 | 0.882 |
| D ³ Net v1 | 27.61 | 0.887 |
| D ³ Net v2 | 28.91 | 0.912 |

D. Results on real image

We tested D³Net on real data captured by LG Nexus 5 mobile phone camera (8 MP, $f/2.4$, 4 mm, 1/6 sec, RGGB). The mobile phone processed the raw data and stored the image as JPEG. We analyzed cropped patch with the size of 449×433 which is shown in Fig 11a. In the inset of the figure, zoomed minipatch of size 46×46 is presented. Notice the demosaicking artifacts in the image.

Intrinsic camera blur kernels for different regions of the input image can be estimated in advance according to [1]. To train our network we artificially blurred test images from Kodak dataset with PSF (Fig. 2b) corresponding to the selected patch of the captured image. We considered additive Gaussian noise 35 dB. Eventually, blurred Kodak images were down-sampled with the Bayer pattern RGGB. In order to prevent over-fitting of the network, size of the restoration and update filters were set to 3×3 . To form training and validation set, Kodak images were cropped into patches of size 210×310 . Other parameters were set as in IV-B. Training of our network lasted approximately 5 minutes on a GeForce RTX 2080 Ti.

The raw image as returned by the camera API was processed through D³Net and the output is seen in Fig 11b. By comparison, the result of our method reveals greater detail, looks visually more pleasing and does not suffer from disturbing demosaicking artifacts. Over-shooting on sharp edges can be adjusted by the fine tuning of the network hyper parameters (especially by the number of the update layers). Small size (3×3) of restoration and update filters makes it particularly suitable for implementation in small embedded system like digital cameras.

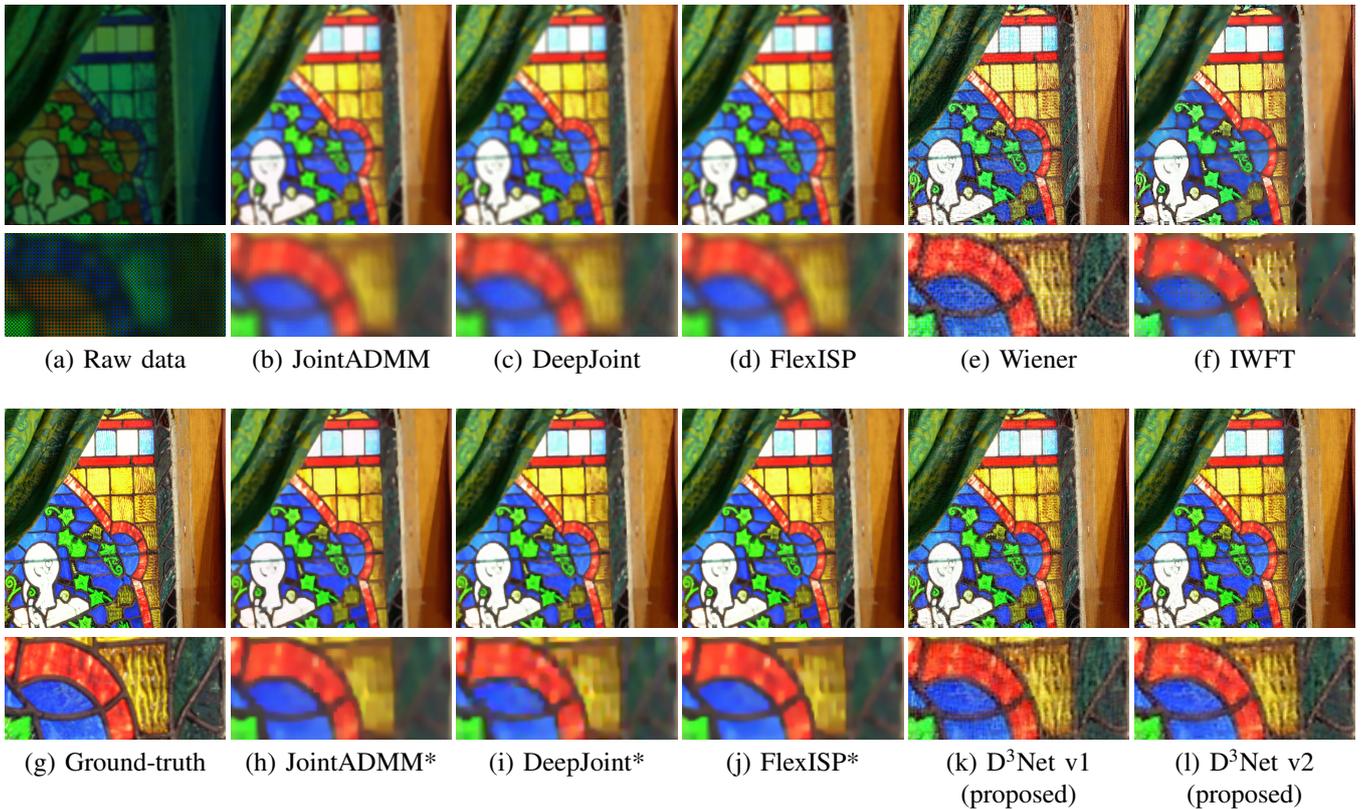


Fig. 10. Comparison of our joint demosaicking deblurring and deringing network D^3Net with sequential demosaicking and deblurring methods (FlexISP [3], DeepJoint [15] and JointADMM [5] followed by robust non-blind deconvolution method [16]). Evaluated on McMaster dataset [14]. Symbol * means that offline deblurring was applied. D^3Net v1 was trained using a single pair and D^3Net v2 was trained on 18 pairs of degraded and ground-truth images from Kodak dataset.



Fig. 11. Image reconstruction of real data captured by LG Nexus 5 phone camera. a) Demosaicking processed by phone, b) D^3Net .

V. CONCLUSIONS

In this work, we presented a novel portable CNN for joint demosaicking, deblurring and deringing of raw image data. The light-weight structure of the network makes it particularly suitable for implementation in digital cameras. Architecture of the proposed network is inspired by the model-based optimization algorithm IWFT. We adopted the IWFT idea, extended it to perform also demosaicking, and designed it as a CNN. Results demonstrate that filters used for image reconstruction can be further improved by adopting the learning-based approach. We have shown, that our joint approach outperforms state-of-the-art demosaicking methods with offline deblurring.

REFERENCES

- [1] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Non-stationary correction of optical aberrations," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 659–666.
- [2] H. Q. Luong, B. Goossens, J. Aelterman, A. Pižurica, and W. Philips, "A primal-dual algorithm for joint demosaicking and deconvolution," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 2801–2804.
- [3] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian *et al.*, "Flexisp: A flexible camera image processing framework," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, pp. 1–13, 2014.

- [4] D. S. Yoo, M. K. Park, and M. G. Kang, "Joint deblurring and demosaicing using edge information from bayer images," *IEICE TRANSACTIONS on Information and Systems*, vol. 97, no. 7, pp. 1872–1884, 2014.
- [5] H. Tan, X. Zeng, S. Lai, Y. Liu, and M. Zhang, "Joint demosaicing and denoising of noisy bayer images with admm," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2951–2955.
- [6] D. S. Tan, W.-Y. Chen, and K.-L. Hua, "Deepdemosaicking: Adaptive image demosaicking via multiple deep fully convolutional networks," *IEEE Transactions on Image Processing*, vol. 27, no. 5, pp. 2408–2419, 2018.
- [7] E. Schwartz, R. Giryes, and A. M. Bronstein, "Deepisp: Toward learning an end-to-end image processing pipeline," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 912–923, 2018.
- [8] F. Kokkinos and S. Lefkimiatis, "Iterative joint image demosaicking and denoising using a residual denoising network," *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4177–4188, 2019.
- [9] F. Šroubek, T. Kerepecký, and J. Kamenický, "Iterative wiener filtering for deconvolution with ringing artifact suppression," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [10] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [11] S. Wang, S. Fidler, and R. Urtasun, "Proximal deep structured models," in *Advances in Neural Information Processing Systems*, 2016, pp. 865–873.
- [12] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.
- [13] H. S. Malvar, L.-w. He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 2004, pp. iii–485.
- [14] L. Zhang, X. Wu, A. Buades, and X. Li, "Color demosaicking by local directional interpolation and nonlocal adaptive thresholding," *Journal of Electronic imaging*, vol. 20, no. 2, p. 023016, 2011.
- [15] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, pp. 1–12, 2016.
- [16] J. Pan, Z. Lin, Z. Su, and M.-H. Yang, "Robust kernel estimation with outliers handling for image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2800–2808.