Deep Image Demosaicking using a Cascade of Convolutional Residual Denoising Networks

Filippos Kokkinos, Stamatis Lefkimmiatis

Skolkovo Institute of Science and Technology, Moscow, Russia

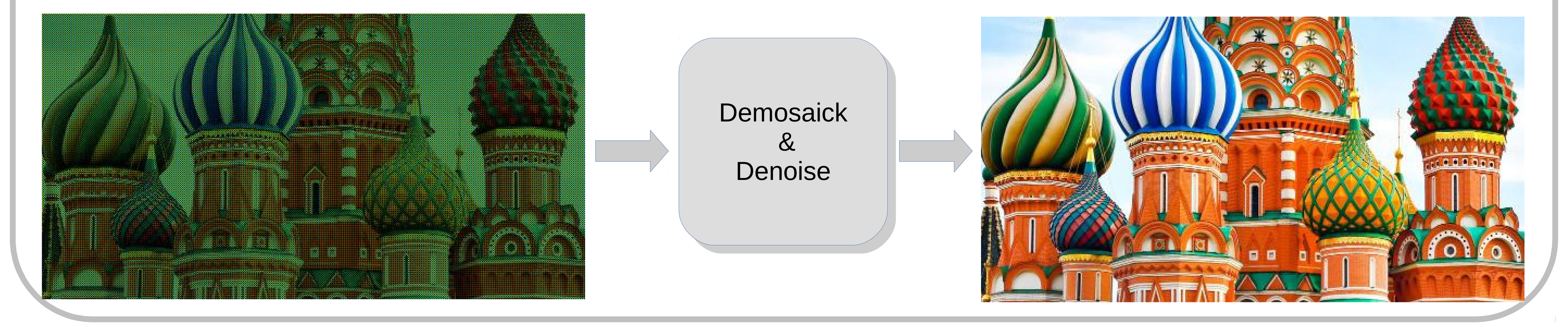
Skoltech — Skolkovo Institute of Science and Technology

ECCV 2018 — European Conference on Computer Vision

1. Motivation - Contributions

- Demosaicking and denoising are among the most crucial steps of modern digital camera pipelines and their joint treatment is a highly ill-posed inverse problem where at-least two-thirds of the information are missing and the rest are corrupted by noise.
  - ✖ Reconstruction errors in the early stages produce unsatisfying end results.
- We designed a novel deep network that
  - is inspired by classical image regularization approaches and optimization strategy.
  - tackles the demosaicking and denoising problem jointly.
  - performs very well even when trained on small datasets and using a small number of parameters.
  - works with any Color Filter Array (CFA).

Noisy Light Intensity Measurements — End Result



2. Problem Formulation

- Demosaicking : estimate $\mathbf{x} \in \mathbb{R}^N$ according to the model : $\mathbf{y} = \mathbf{M}\mathbf{x} + \mathbf{n}$
  - $\mathbf{M} \in \mathbb{R}^{N \times N}$ is a diagonal binary and singular matrix that models the CFA pattern

degradation matrix — measurements — gaussian i.i.d noise

- Variational problem formulation (P1):

$$\boxed{Q(\mathbf{x}) = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2^2 + \phi(\mathbf{x})}$$

regularizer

- Two challenges arise, how to:
  - Efficiently minimize $Q(\mathbf{x})$ function
  - Properly define $\phi(\mathbf{x})$ to constrain the set of solutions
- We opt to design a network that successfully overcomes both challenges

3. Majorization Minimization (MM) Framework

- Minimize (P1) via successive minimization of a surrogate/majorizer function $\tilde{Q}(\mathbf{x}, \mathbf{x}^{(t)})$ that upper-bounds (P1)
- Iteratively solve $\mathbf{x}^* = \arg\min_x Q(\mathbf{x})$ via $\mathbf{x}^{(t+1)} = \arg\min_x \tilde{Q}(\mathbf{x}; \mathbf{x}^{(t)})$
- Where $\boxed{\tilde{Q}(\mathbf{x}; \mathbf{x}^{(t)}) > Q(\mathbf{x}), \forall \mathbf{x} \neq \mathbf{x}^{(t)}}$ and $\boxed{\tilde{Q}(\mathbf{x}^{(t)}; \mathbf{x}^{(t)}) = Q(\mathbf{x}^{(t)})}$
- Deriving a majorizer of the data-fidelity term:

$$\tilde{d}(\mathbf{x}, \mathbf{x}_0) = \frac{1}{2\sigma^2}\|\mathbf{y} - \mathbf{M}\mathbf{x}\|_2^2 + d(\mathbf{x}, \mathbf{x}_0)$$

  - Where $d(\mathbf{x}, \mathbf{x}_0) = \frac{1}{2\sigma^2}(\mathbf{x} - \mathbf{x}_0)^T[\alpha\mathbf{I} - \mathbf{M}](\mathbf{x} - \mathbf{x}_0)$ is a distance function between $\mathbf{x}$ and $\mathbf{x}_0$
  - For requirements to hold, $d(\mathbf{x}, \mathbf{x}_0)$ needs to be PSD matrix therefore $\alpha > \|\mathbf{M}\|_2 \geq 1$
- The majorizer take the following form that resembles a denoising problem (P2):

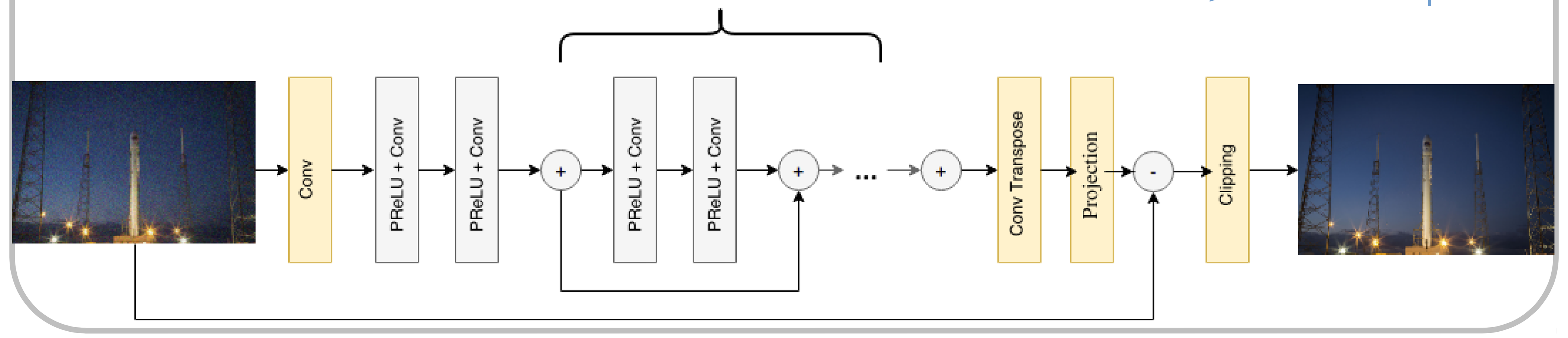
$$\tilde{Q}(\mathbf{x}, \mathbf{x}_0) = \frac{1}{2(\sigma/\sqrt{a})^2}\|\mathbf{x} - \mathbf{z}\|_2^2 + \phi(\mathbf{x}) + c, \text{ where } \mathbf{z} = \mathbf{y} + (\mathbf{I} - \mathbf{M})\mathbf{x}_0$$

4. Residual Denoising Network (ResDNet)

- We need to derive a network that works for a wide range of noise levels to solve (P2)
  - Make use of *D* CNNs layers with weight normalization and PReLU activation function
  - Implement a residual approach to subtract the noise realization
- Projection Layer normalizes the noise realization estimate to have the desired variance σ .

$$\mathcal{P}_{\mathcal{C}}(\mathbf{y}) = \varepsilon \frac{\mathbf{y}}{max(\|\mathbf{y}\|_2, \varepsilon)}, \ \varepsilon = e^{\gamma}\theta, \ \theta = \sigma\sqrt{N-1}$$

Number of pixels



5. Demosaicking Network Architecture

- Unroll *K* Majorization Minimization iterations as a network and use *ResDNet* for each denoising step
  - Promote parameter sharing in each step to keep a low number of parameters (~380.000)
  - Extrapolate results of each iteration to accelerate MM Framework

Algorithm 1: The proposed demosaicking network described as an iterative process. The ResDnet parameters remain the same in every iteration.

Input: $\mathbf{M}$: CFA, $\mathbf{y}$: input, K : iterations, $\boldsymbol{w} \in \mathbb{R}^K$: extrapolation weights, $\boldsymbol{\sigma} \in \mathbb{R}^K$: noise vector

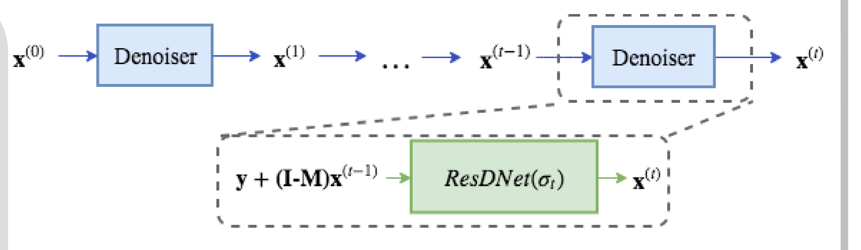
$\mathbf{x}^0 = \mathbf{0}$, $\mathbf{x}^1 = \mathbf{y}$;

for $i \leftarrow 1$ **to** K **do**

  $\mathbf{u} = \mathbf{x}^{(i)} + \boldsymbol{w}_i(\mathbf{x}^{(i)} - \mathbf{x}^{(i-1)})$;

  $\mathbf{x}^{(i+1)} = \mathrm{ResDNet}((\mathbf{I} - \mathbf{M})\mathbf{u} + \mathbf{y}, \boldsymbol{\sigma}_i)$;

end

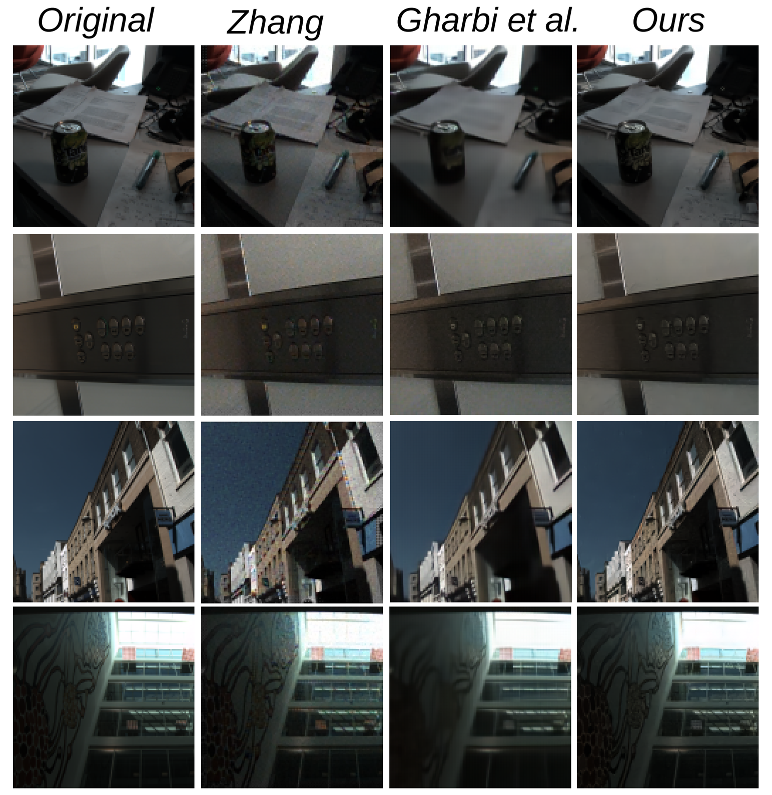


6. Comparisons & Results

- Bayer Demosaicking PSNR performance evaluation:

	noise-free linRGB	noise-free sRGB	noisy linRGB	noisy sRGB
Non-ML Methods:				
bilinear	30.9	24.9	-	-
Zhang(NLM) [2]	38.4	32.1	-	-
Getreuer [41]	39.4	32.9	-	-
Heide [5]	40.0	33.8	-	-
Trained on MSR Dataset:				
Khasabi [14]	39.4	32.6	37.8	31.5
Klatzer [19]	40.9	34.6	38.8	32.6
Bigdeli [42]	-	-	38.7	-
ours	**41.0**	**34.6**	**39.2**	**33.3**
Trained on MIT Dataset:				
Gharbi (sRGB)[20]	41.6	35.3	38.4	32.5
Gharbi (linRGB)	**42.7**	**35.9**	38.6	32.6
ours* (linRGB)	42.6	**35.9**	N/A	N/A

Original — Zhang — Gharbi et al. — Ours



- Evaluation an simulated data:

	Kodak	McM	Vdp	Moire
Non-ML Methods:				
bilinear	32.9	32.5	25.2	27.6
Adobe Camera Raw 9	33.9	32.2	27.8	29.8
Buades [4]	37.3	35.5	29.7	31.7
Zhang (NLM) [2]	37.9	36.3	30.1	31.9
Getreuer [41]	38.1	36.1	30.8	32.5
Heide [5]	40.0	38.6	27.1	34.9
Trained on MSR Dataset:				
Klatzer [19]	35.3	30.8	28.0	30.3
ours	39.2	34.1	29.2	29.7
Trained on MIT Dataset:				
Gharbi [20]	41.2	39.5	34.3	**37.0**
ours*	**41.5**	**39.7**	**34.5**	**37.0**

- Fuji XTRANS Demosaicking evaluation:

	noise-free linear	noise-free sRGB
Trained on MSR Dataset:		
Khashabi [14]	36.9	30.6
Klatzer [19]	39.6	33.1
ours	**39.9**	**33.7**
Trained on MIT Dataset:		
Gharbi [20]	39.7	33.2