Iterative Residual Networks for Burst Photography Applications

Stamatis Lefkimmiatis

Skolkovo Institute of Science and Technology

Burst Photography

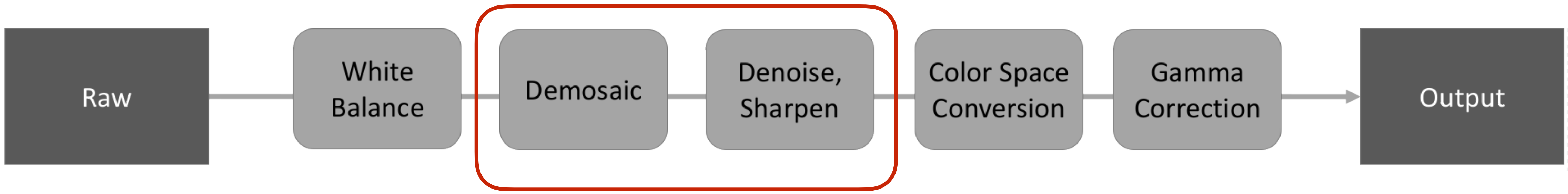
- Nowadays smartphone cameras dominate the photography market (+ 1 Billion smartphone devices are sold every year)
- Mobile cameras are small and versatile but suffer from several hardware constraints (lack of aperture lenses, small sensors with fewer photodiodes)
- As a result image quality cannot match that of DSLR cameras
- To compensate for the hardware limitations focus is shifted towards the software of the camera.
- Burst photography : Fusion of several low-quality frames to produce a single high-quality image

Burst of raw frames

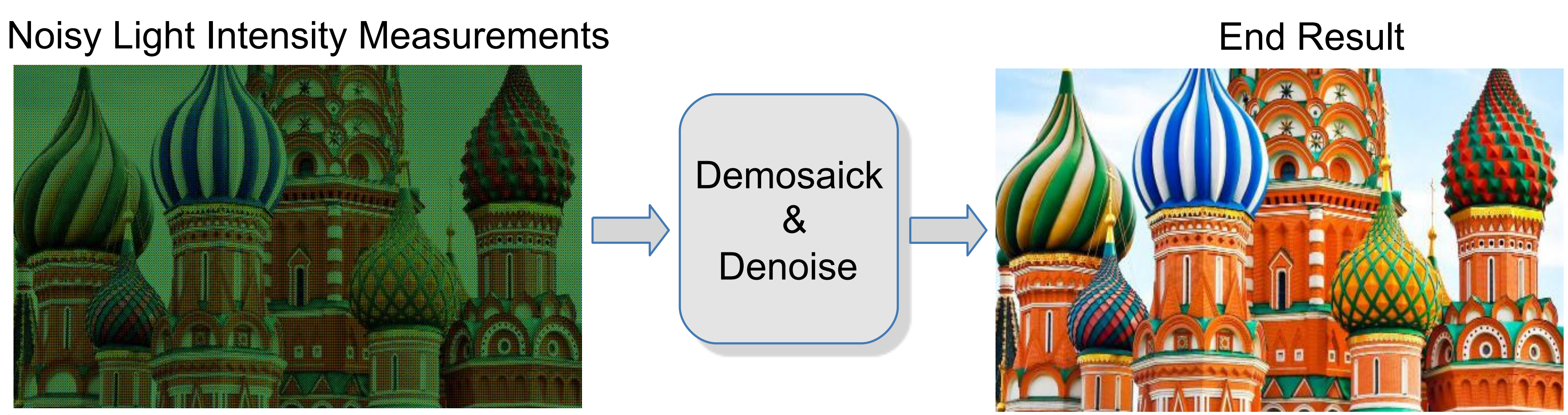
Final high-quality result

Camera Image-Signal Processing Pipeline

- First steps in the camera ISP :
  - Conversion of intensity readings to color images



- **Challenging** problem
  - Only one third of the intensities are known
  - Measurements are further distorted by noise



Problem Formulation

- Forward model : Image acquisition can be accurately described as

measurement noise

burst sequence $\longleftarrow$ $\mathbf{y} = \tilde{\mathbf{H}} S\left(\mathbf{x}\right) + \mathbf{n}$

camera response

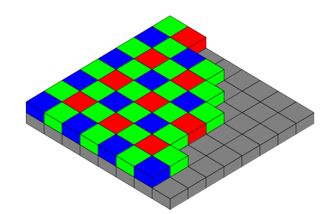
underlying color image

$$\mathbf{y} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_{\boxed{B}} \end{bmatrix} \qquad \tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{H} & & \mathbf{0} \\ \vdots & & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{H} \end{bmatrix} \qquad S\left(\mathbf{x}\right) = \begin{bmatrix} S_1\left(\mathbf{x}\right) \\ S_2\left(\mathbf{x}\right) \\ \vdots \\ S_B\left(\mathbf{x}\right) \end{bmatrix} \qquad \mathbf{n} = \begin{bmatrix} \mathbf{n}_1 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{n}_B \end{bmatrix}$$

burst size

$\mathbf{H}$: problem specific linear operator

- $\mathbf{H} = \mathbf{I}$ (burst Gaussian denoising)
- $\mathbf{H}$: **CFA pattern** (burst demosaicking)



$S_i : \mathbb{R}^N \mapsto \mathbb{R}^N$ affine transformation of the image coordinates (**rotation + translation**)

- Useful observation : Given the warping matrix (estimated by the ECC method) the operator S_i corresponds to a linear interpolation (e.g bilinear) and can be linearized

$$S_i\left(\mathbf{x}\right) \equiv \mathbf{S}_i \mathbf{x}$$

Variational / Bayesian Approach : Overview

measurement noise

burst sequence $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$

camera response underlying color image

$$\mathbf{A} = \tilde{\mathbf{H}}\mathbf{S}$$

- Burst Image restoration
  - Ill-posed problem
  - Unique solution **does not exist**

Variational / Bayesian Approach : Overview

measurement noise

burst sequence ⟵ $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$

camera response underlying color image

$$\mathbf{A} = \tilde{\mathbf{H}}\mathbf{S}$$

- Burst Image restoration
  - Ill-posed problem
  - Unique solution **does not exist**
- Variational framework - Energy Minimization

data fidelity

$$E\left(\mathbf{x}\right) = d\left(\mathbf{x};\mathbf{A},\mathbf{y}\right) + \lambda \cdot r\left(\mathbf{x}\right)$$ regularizer

Variational / Bayesian Approach : Overview

measurement noise

burst sequence $\longleftarrow \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$ $\qquad \mathbf{A} = \tilde{\mathbf{H}}\mathbf{S}$

camera response underlying color image

- Burst Image restoration
  - Ill-posed problem
  - Unique solution **does not exist**
- Variational framework - Energy Minimization

data fidelity

$$E(\mathbf{x}) = d(\mathbf{x}; \mathbf{A}, \mathbf{y}) + \lambda \cdot r(\mathbf{x})$$ regularizer

- Direct relation to Bayesian methods :
  - data fidelity $\Longrightarrow$ observation log-likelihood
  - regularizer $\Longrightarrow$ prior distribution
  - minimizer $\Longrightarrow$ MAP estimate

Image Regularization

- Regularizer : key component of the reconstruction
  - Constrains the set of plausible solutions
  - Prior knowledge of desirable image properties

Image Regularization

- Regularizer : key component of the reconstruction
  - Constrains the set of plausible solutions
  - Prior knowledge of desirable image properties

- Generic form of regularization functionals :

$$r(\mathbf{x}) = \sum_{k=1}^{K} \phi(\mathbf{L}_k \mathbf{x})$$

potential function

regularization operator

  - $\mathbf{L} : \mathbb{R}^N \mapsto \mathbb{R}^{K \times D}$, linear operator (gradient, Hessian, wavelets, …)
  - ϕ : ℓ_2 squared norm (Tikhonov-Miller regularization) , ℓ_1-norm, log function, pseudo ℓ_0-norm, Schatten-norms, etc …

- Quadratic regularizers
  - mathematical simplicity
  - computational tractability
- Non-quadratic regularizers
  - less sensitive to outliers
  - improved reconstruction results

Total Variation

- One of the most popular regularizers for imaging applications [ROF `92]
  - Provides a global measure of the image intensity variations

Continuous-domain definition

$$\mathrm{TV}(u) = \int_{\Omega} \|\nabla u(\boldsymbol{r})\|_2 \, d\boldsymbol{r}$$

Discrete-domain definition

$$\mathrm{TV}(\mathbf{u}) = \sum_{n=1}^{N} \|(\nabla \mathbf{u})_n\|_2$$

- Numerous applications in image processing and computer vision
  - ✔ Invariance properties w.r.t transformations of the coordinate system
  - ✔ L_1 type of behavior that does not over-penalize large intensity variations
  - ✔ Reconstructed images with sharp edges
  - ✔ Convex functional - Efficient minimization techniques
  - ✗ Promotes piecewise constant reconstructions
  - ✗ Presence of staircase artifacts in smooth regions

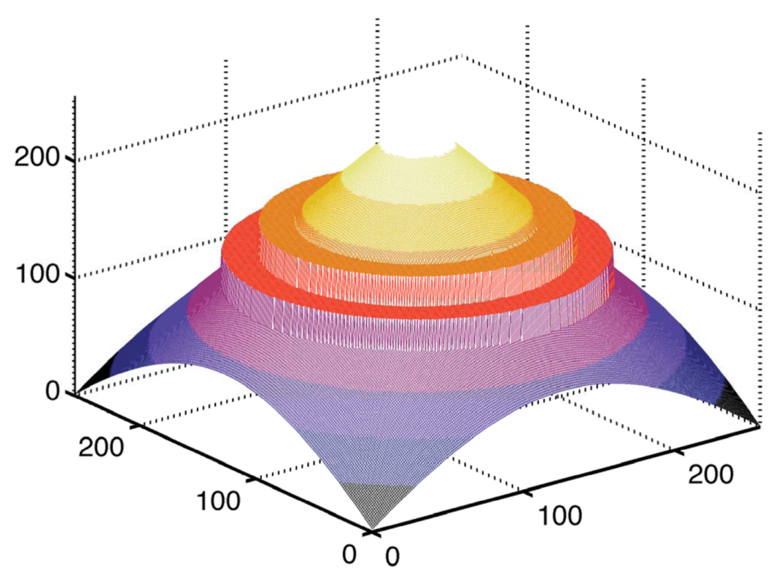
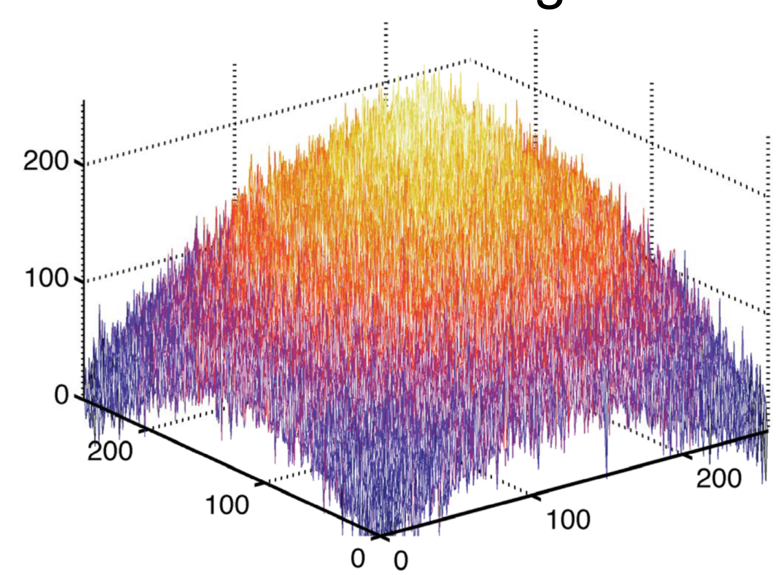
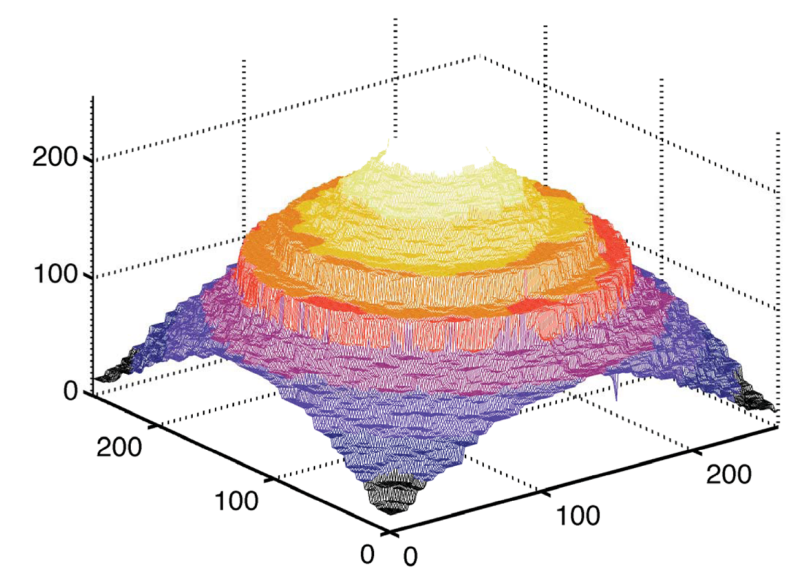


Image intensity profile



Noisy intensity profile



TV reconstruction

Machine Learning for Inverse Problems

- Two major challenges under the variational / minimization framework
  - Specify the exact form of the regularization operator and potential function
  - Recover the solution using an iterative approach which possibly requires a large number of updates

Machine Learning for Inverse Problems

- Two major challenges under the variational / minimization framework
  - Specify the exact form of the regularization operator and potential function
  - Recover the solution using an iterative approach which possibly requires a large number of updates
- Machine Learning alternative
  - Design a network with the capacity to learn the involved quantities from training data
  - Discriminative learning by using a loss-minimization strategy

Machine Learning for Inverse Problems

- Two major challenges under the variational / minimization framework
  - Specify the exact form of the regularization operator and potential function
  - Recover the solution using an iterative approach which possibly requires a large number of updates
- Machine Learning alternative
  - Design a network with the capacity to learn the involved quantities from training data
  - Discriminative learning by using a loss-minimization strategy
- Main challenge for ML
  - How to choose a proper architecture for the graph of the network?

Machine Learning for Inverse Problems

- Two major challenges under the variational / minimization framework
  - Specify the exact form of the regularization operator and potential function
  - Recover the solution using an iterative approach which possibly requires a large number of updates
- Machine Learning alternative
  - Design a network with the capacity to learn the involved quantities from training data
  - Discriminative learning by using a loss-minimization strategy
- Main challenge for ML
  - How to choose a proper architecture for the graph of the network?
- Principled approach for defining the network graph
  - Unroll an iterative optimization method
  - Use a limited number of updates to construct the network
  - Parametrize the regularization operator and the potential function

Variational Approach Revisited

measurement noise

burst sequence $\leftarrow$ $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{n}$

camera response underlying color image

- Image restoration as the solution of a minimization problem

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \underbrace{\frac{1}{2\sigma^2 B} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + r(\mathbf{x})}_{Q(\mathbf{x})}$$

- Challenges to deal with :
  - coupling of the camera's impulse response with the solution
  - non-trivial way to minimize the objective function
  - choice of an appropriate regularizer

Majorization-Minimization framework

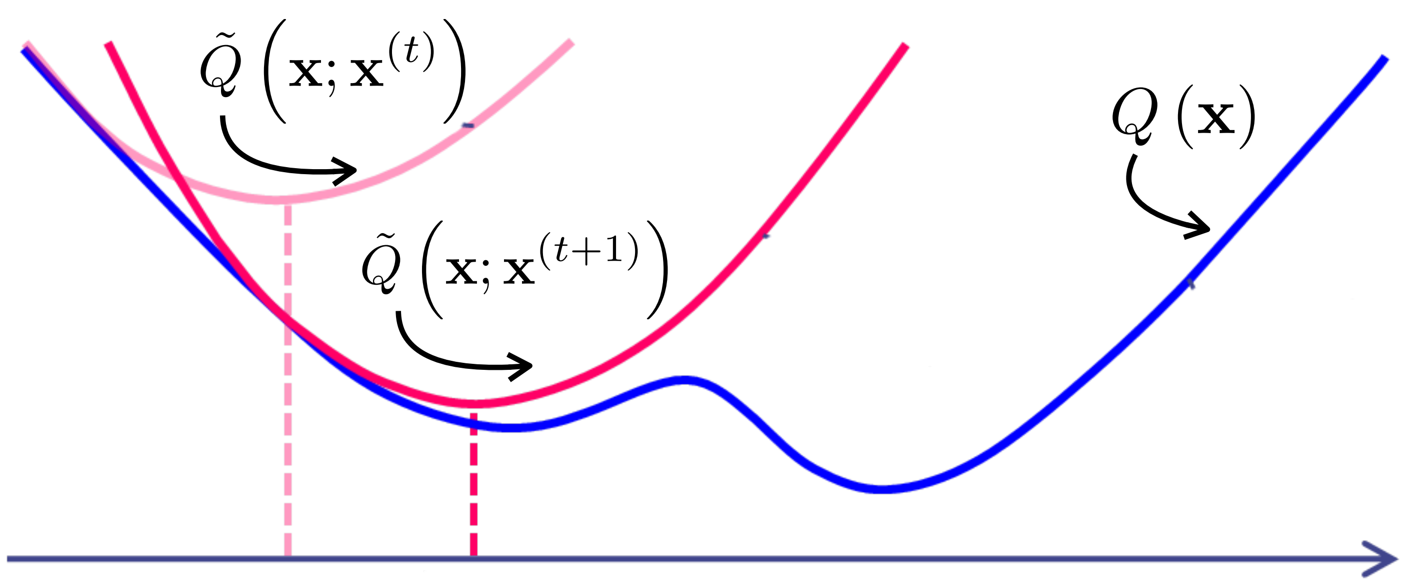
- Iterative minimization approach of the form :

$$\mathbf{x}^{(t+1)} = \arg\min_{\mathbf{x}} \tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right)$$

- $\tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right)$ is a majorizer of the objective function that satisfies the conditions:

$$(\alpha) \quad \tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right) > Q\left(\mathbf{x}\right), \forall \mathbf{x} \neq \mathbf{x}^{(t)}$$

$$(\beta) \quad \tilde{Q}\left(\mathbf{x}^{(t)};\mathbf{x}^{(t)}\right) = Q\left(\mathbf{x}^{(t)}\right)$$



MM for Burst Image Restoration

- Derive a majorizer by upper-bounding the data fidelity term

$$\tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \underbrace{\frac{1}{2\sigma^2 B}\left\|\mathbf{y}-\mathbf{A}\mathbf{x}\right\|_2^2}_{d(\mathbf{x})} + g\left(\mathbf{x},\mathbf{x}^{(t)}\right)$$

$$g\left(\mathbf{x},\mathbf{x}^{(t)}\right) = \frac{1}{2\sigma^2 B}\left(\mathbf{x}-\mathbf{x}^{(t)}\right)^T\left[\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}\right]\left(\mathbf{x}-\mathbf{x}^{(t)}\right)$$

MM for Burst Image Restoration

- Derive a majorizer by upper-bounding the data fidelity term

$$\tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \underbrace{\frac{1}{2\sigma^2 B}\left\|\mathbf{y}-\mathbf{A}\mathbf{x}\right\|_2^2}_{d(\mathbf{x})} + g\left(\mathbf{x},\mathbf{x}^{(t)}\right)$$

$$g\left(\mathbf{x},\mathbf{x}^{(t)}\right) = \frac{1}{2\sigma^2 B}\left(\mathbf{x}-\mathbf{x}^{(t)}\right)^T\left[\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}\right]\left(\mathbf{x}-\mathbf{x}^{(t)}\right)$$

- $\tilde{d}\left(\cdot,\cdot\right)$ is a valid majorizer iff $g\left(\mathbf{x},\mathbf{y}\right) > 0\,, \forall \mathbf{x} \neq \mathbf{y}$ and $g\left(\mathbf{x},\mathbf{x}\right) = 0$
  - $\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}$ is positive definite $\Rightarrow\ \alpha > \left\|\mathbf{A}^T\mathbf{A}\right\|$

MM for Burst Image Restoration

- Derive a majorizer by upper-bounding the data fidelity term

$$\tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \underbrace{\frac{1}{2\sigma^2 B}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2}_{d(\mathbf{x})} + g\left(\mathbf{x},\mathbf{x}^{(t)}\right)$$

$$g\left(\mathbf{x},\mathbf{x}^{(t)}\right) = \frac{1}{2\sigma^2 B}\left(\mathbf{x}-\mathbf{x}^{(t)}\right)^T\left[\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}\right]\left(\mathbf{x}-\mathbf{x}^{(t)}\right)$$

- $\tilde{d}(\cdot,\cdot)$ is a valid majorizer iff $g(\mathbf{x},\mathbf{y}) > 0\,, \forall \mathbf{x} \neq \mathbf{y}$ and $g(\mathbf{x},\mathbf{x}) = 0$
  - $\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}$ is positive definite $\Rightarrow\ \alpha > \left\|\mathbf{A}^T\mathbf{A}\right\|$

- Overall majorizer is of the form :

$$\begin{aligned}\tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right) &= \tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) + r(\mathbf{x}) \\ &= \frac{\alpha}{2\sigma^2 B}\|\mathbf{x}-\mathbf{z}\|_2^2 + r(\mathbf{x}) + \text{const.}\end{aligned}$$

$$\mathbf{z} = \mathbf{x}^{(t)} + \frac{1}{\alpha}\sum_{i=1}^{B}\mathbf{S}_i^T\mathbf{H}^T\left(\mathbf{y}_i - \mathbf{H}\mathbf{S}_i\mathbf{x}^{(t)}\right)$$

MM for Burst Image Restoration

- Derive a majorizer by upper-bounding the data fidelity term

$$\tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \underbrace{\frac{1}{2\sigma^2 B}\left\|\mathbf{y}-\mathbf{A}\mathbf{x}\right\|_2^2}_{d(\mathbf{x})} + g\left(\mathbf{x},\mathbf{x}^{(t)}\right)$$

$$g\left(\mathbf{x},\mathbf{x}^{(t)}\right) = \frac{1}{2\sigma^2 B}\left(\mathbf{x}-\mathbf{x}^{(t)}\right)^T\left[\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}\right]\left(\mathbf{x}-\mathbf{x}^{(t)}\right)$$

- $\tilde{d}\left(\cdot,\cdot\right)$ is a valid majorizer iff $g\left(\mathbf{x},\mathbf{y}\right)>0\,,\forall\mathbf{x}\neq\mathbf{y}$ and $g\left(\mathbf{x},\mathbf{x}\right)=0$
  - $\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}$ is positive definite $\Rightarrow\ \alpha > \left\|\mathbf{A}^T\mathbf{A}\right\|$

- Overall majorizer is of the form :

$$\tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) + r\left(\mathbf{x}\right)$$

denoising objective function

$$= \boxed{\frac{\alpha}{2\sigma^2 B}\left\|\mathbf{x}-\mathbf{z}\right\|_2^2 + r\left(\mathbf{x}\right) + \text{const.}}$$

$$\mathbf{z} = \mathbf{x}^{(t)} + \frac{1}{\alpha}\sum_{i=1}^{B}\mathbf{S}_i^T\mathbf{H}^T\left(\mathbf{y}_i-\mathbf{H}\mathbf{S}_i\mathbf{x}^{(t)}\right)$$

MM-based Burst Image Restoration

- Derive a majorizer by upper-bounding the data fidelity term

$$\tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \underbrace{\frac{1}{2\sigma^2 B}\|\mathbf{y}-\mathbf{A}\mathbf{x}\|_2^2}_{d(\mathbf{x})} + g\left(\mathbf{x},\mathbf{x}^{(t)}\right)$$

$$g\left(\mathbf{x},\mathbf{x}^{(t)}\right) = \frac{1}{2\sigma^2 B}\left(\mathbf{x}-\mathbf{x}^{(t)}\right)^T\left[\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}\right]\left(\mathbf{x}-\mathbf{x}^{(t)}\right)$$

- $\tilde{d}(\cdot,\cdot)$ is a valid majorizer iff $g(\mathbf{x},\mathbf{y}) > 0\,,\forall\mathbf{x}\neq\mathbf{y}$ and $g(\mathbf{x},\mathbf{x}) = 0$
  - $\alpha\mathbf{I}-\mathbf{A}^T\mathbf{A}$ is positive definite $\Rightarrow\ \alpha > \left\|\mathbf{A}^T\mathbf{A}\right\| \Rightarrow \boxed{\alpha \geq B}$

- Overall majorizer is of the form :

$$\tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right) = \tilde{d}\left(\mathbf{x};\mathbf{x}^{(t)}\right) + r(\mathbf{x})$$

denoising objective function

$$= \boxed{\frac{1}{2\sigma^2}\|\mathbf{x}-\mathbf{z}\|_2^2 + r(\mathbf{x}) + \text{const.}}$$

$$\mathbf{z} = \mathbf{x}^{(t)} + \frac{1}{B}\sum_{i=1}^{B}\mathbf{S}_i^T\mathbf{H}^T\left(\mathbf{y}_i - \mathbf{H}\mathbf{S}_i\mathbf{x}^{(t)}\right)$$

Image Denoising Networks

- Image denoising solution (P1)

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right)$$

Image Denoising Networks

- Image denoising solution (P1)

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right)$$

- Regularization parameter λ : “free” parameter
  - Different values lead to different restoration results of varying image quality
  - **No direct way** to *a priori* relate the value of λ with the restoration quality
  - Networks derived based on this formulation are noise-level specific [Schmidt & Roth `14, Chen et al. `16, Lefkimmiatis `17]
    - A new set of parameters must be learned for every noise level

Image Denoising Networks

- Image denoising solution (P1)

$$\mathbf{x}^\star = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \lambda \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right)$$

- Regularization parameter λ : “free” parameter
  - Different values lead to different restoration results of varying image quality
  - **No direct way** to *a priori* relate the value of λ with the restoration quality
  - Networks derived based on this formulation are noise-level specific [Schmidt & Roth `14, Chen et al. `16, Lefkimmiatis `17]
    - A new set of parameters must be learned for every noise level
- Alternative way : Constrained problem formulation (P2)

$$\mathbf{x}^\star = \arg\min_{\|\mathbf{y}-\mathbf{x}\|_2 \le \varepsilon} \left(\sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) \equiv r\left(\mathbf{x}\right) \right)$$

  - $\|\mathbf{y}-\mathbf{x}\|_2 = \|\mathbf{n}\|_2 \propto \sigma$
  - Free parameter ε is easier to tune - directly related to the noise std
  - Several available methods for estimating the noise level from observed data

Lefkimmiatis CVPR `17 & `18

Image Denoising Networks

- Image denoising solution (P1)

$$\mathbf{x}^{\star} = \arg\min_{\mathbf{x}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{x}\right\|_2^2 + \lambda \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right)$$

- Regularization parameter λ : "free" parameter
  - Different values lead to different restoration results of varying image quality
  - **No direct way** to *a priori* relate the value of λ with the restoration quality
  - Networks derived based on this formulation are noise-level specific [Schmidt & Roth \`14, Chen et al. \`16, Lefkimmiatis \`17]
    - A new set of parameters must be learned for every noise level
- Alternative way : Constrained problem formulation (P2)

$$\mathbf{x}^{\star} = \arg\min_{\left\|\mathbf{y}-\mathbf{x}\right\|_2 \le \varepsilon} \left(\sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) \equiv r\left(\mathbf{x}\right)\right)$$

  - $\left\|\mathbf{y}-\mathbf{x}\right\|_2 = \left\|\mathbf{n}\right\|_2 \propto \sigma$
  - Free parameter ε is easier to tune - directly related to the noise std
  - Several available methods for estimating the noise level from observed data
- The two problems are equivalent in the following sense :
  - For any $\varepsilon > 0$ such that P2 is feasible, the solution of P2 is either the null vector or else it is a solution of P1 for some $\lambda > 0$

Lefkimmiatis CVPR \`17 & \`18

Proximal Gradient Method

- Unconstrained form of P2

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

indicator function $\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right) = \begin{cases} 0, & \text{if } \left\|\mathbf{y}-\mathbf{x}\right\|_2 \leq \varepsilon \\ \infty, & \text{otherwise} \end{cases}$

Proximal Gradient Method

- Unconstrained form of P2

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

indicator function

$$\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right) = \begin{cases} 0, & \text{if } \left\|\mathbf{y}-\mathbf{x}\right\|_2 \le \varepsilon \\ \infty, & \text{otherwise} \end{cases}$$

- PGM : gradient-descent method that can handle non-smooth functionals

Proximal Gradient Method

- Unconstrained form of P2

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

indicator function
$$\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right) = \begin{cases} 0, & \text{if } \|\mathbf{y}-\mathbf{x}\|_2 \leq \varepsilon \\ \infty, & \text{otherwise} \end{cases}$$

- PGM : gradient-descent method that can handle non-smooth functionals
  - The objective is split into a smooth and a non-smooth part
  - Must be able to compute the proximal operator of the non-smooth functional

$$\mathrm{prox}_{\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}}\left(\mathbf{z}\right) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|_2^2 + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

Proximal Gradient Method

- Unconstrained form of P2

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \sum_{k=1}^{K} \phi\left(\mathbf{L}_k \mathbf{x}\right) + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

indicator function

$$\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right) = \begin{cases} 0, & \text{if } \|\mathbf{y}-\mathbf{x}\|_2 \leq \varepsilon \\ \infty, & \text{otherwise} \end{cases}$$

- PGM : gradient-descent method that can handle non-smooth functionals
  - The objective is split into a smooth and a non-smooth part
  - Must be able to compute the proximal operator of the non-smooth functional

$$\operatorname{prox}_{\iota_{\mathcal{C}(\mathbf{y},\varepsilon)}}\left(\mathbf{z}\right) = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{x}-\mathbf{z}\|_2^2 + \iota_{\mathcal{C}(\mathbf{y},\varepsilon)}\left(\mathbf{x}\right)$$

- Updates of the solution using PGM (assuming a smooth regularizer)

adaptive step size

$$\boxed{\mathbf{x}^t = \operatorname{prox}_{\gamma^t \iota_C}\left(\mathbf{x}^{t-1} - \gamma^t \nabla_{\mathbf{x}} r\left(\mathbf{x}^{t-1}\right)\right)}$$

Proximal Gradient Method Cont'd

- Gradient of the smooth part

$$\nabla_{\mathbf{x}} r\left(\mathbf{x}\right) = \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi\left(\mathbf{L}_k \mathbf{x}\right) \equiv h\left(\mathbf{x}\right)$$

$$\psi\left(\mathbf{z}\right) = \nabla_{\mathbf{z}} \phi\left(\mathbf{z}\right), \ \mathbf{z} \in \mathbb{R}^D$$

Proximal Gradient Method Cont'd

- Gradient of the smooth part

$$\nabla_{\mathbf{x}} r(\mathbf{x}) = \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi(\mathbf{L}_k \mathbf{x}) \equiv h(\mathbf{x})$$

$$\psi(\mathbf{z}) = \nabla_{\mathbf{z}} \phi(\mathbf{z}), \ \mathbf{z} \in \mathbb{R}^D$$

- Proximal map of the non-smooth part
  - Orthogonal projection onto the set $\mathcal{C}(\mathbf{y}, \varepsilon)$

$$\Pi_{\mathcal{C}}(\mathbf{z}) = \underset{\mathbf{x} \in \mathcal{C}(\mathbf{y}, \varepsilon)}{\arg\min} \|\mathbf{x} - \mathbf{z}\|_2^2$$

Proximal Gradient Method Cont'd

- Gradient of the smooth part

$$\nabla_{\mathbf{x}} r\left(\mathbf{x}\right) = \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi\left(\mathbf{L}_k \mathbf{x}\right) \equiv h\left(\mathbf{x}\right)$$

$$\psi\left(\mathbf{z}\right) = \nabla_{\mathbf{z}} \phi\left(\mathbf{z}\right), \ \mathbf{z} \in \mathbb{R}^D$$

- Proximal map of the non-smooth part
  - Orthogonal projection onto the set $\mathcal{C}\left(\mathbf{y}, \varepsilon\right)$

$$\Pi_{\mathcal{C}}\left(\mathbf{z}\right) = \underset{\mathbf{x} \in \mathcal{C}(\mathbf{y}, \varepsilon)}{\arg\min} \left\|\mathbf{x} - \mathbf{z}\right\|_2^2$$

  - An analytical solution **exists**

$$\Pi_{\mathcal{C}}\left(\boldsymbol{z}\right) = \mathbf{y} + \varepsilon \frac{\boldsymbol{z} - \mathbf{y}}{\max\left(\left\|\boldsymbol{z} - \mathbf{y}\right\|_2, \varepsilon\right)}$$

Proximal Gradient Method Cont'd

- Gradient of the smooth part

$$\nabla_{\mathbf{x}} r(\mathbf{x}) = \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi(\mathbf{L}_k \mathbf{x}) \equiv h(\mathbf{x})$$

$$\psi(\mathbf{z}) = \nabla_{\mathbf{z}} \phi(\mathbf{z}), \; \mathbf{z} \in \mathbb{R}^D$$

- Proximal map of the non-smooth part
  - Orthogonal projection onto the set $\mathcal{C}(\mathbf{y}, \varepsilon)$

$$\Pi_{\mathcal{C}}(\mathbf{z}) = \underset{\mathbf{x} \in \mathcal{C}(\mathbf{y}, \varepsilon)}{\arg\min} \|\mathbf{x} - \mathbf{z}\|_2^2$$

  - An analytical solution **exists**

$$\Pi_{\mathcal{C}}(\boldsymbol{z}) = \mathbf{y} + \varepsilon \frac{\boldsymbol{z} - \mathbf{y}}{\max\left(\|\boldsymbol{z} - \mathbf{y}\|_2, \varepsilon\right)}$$

- Proximal gradient iterations

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^t\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^t(\mathbf{x}) = \gamma^t h(\mathbf{x})$$

Normalized residual iterations

- PGM updates

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^t\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^t\left(\mathbf{x}\right) = \gamma^t\, \nabla_{\mathbf{x}} r\left(\mathbf{x}\right)$$

Normalized residual iterations

- PGM updates

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^t\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^t(\mathbf{x}) = \gamma^t \nabla_{\mathbf{x}} r(\mathbf{x})$$

- Interpretation : Iterative residual denoising
  - The solution is obtained by recursively subtracting from the input refined estimates of the noise realization distorting the input

Normalized residual iterations

- PGM updates

$$\mathbf{x}^{t} = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^{t}\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^{t}\left(\mathbf{x}\right) = \gamma^{t}\,\nabla_{\mathbf{x}} r\left(\mathbf{x}\right)$$

- Interpretation : Iterative residual denoising
  - The solution is obtained by recursively subtracting from the input refined estimates of the noise realization distorting the input

$$\mathbf{x}^{1} = \mathbf{y} - \varepsilon \frac{h^{1}\left(\mathbf{y}\right)}{\max\left(\left\|h^{1}\left(\mathbf{y}\right)\right\|_{2}, \varepsilon\right)} = \mathbf{x} + \left(\mathbf{n} - \mathbf{n}^{1}\right)$$

Normalized residual iterations

- PGM updates

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^t\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^t\left(\mathbf{x}\right) = \gamma^t \nabla_{\mathbf{x}} r\left(\mathbf{x}\right)$$

- Interpretation : Iterative residual denoising
  - The solution is obtained by recursively subtracting from the input refined estimates of the noise realization distorting the input

$$\mathbf{x}^1 = \mathbf{y} - \varepsilon \frac{h^1\left(\mathbf{y}\right)}{\max\left(\left\|h^1\left(\mathbf{y}\right)\right\|_2, \varepsilon\right)} = \mathbf{x} + \left(\mathbf{n} - \mathbf{n}^1\right)$$

$$\vdots$$

$$\mathbf{x}^k = \mathbf{y} - \mathbf{n}^k = \mathbf{x} + \left(\mathbf{n} - \mathbf{n}^k\right), \quad k > 1$$

Normalized residual iterations

- PGM updates

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - h^t\left(\mathbf{x}^{t-1}\right)\right), \text{ where } h^t(\mathbf{x}) = \gamma^t \nabla_{\mathbf{x}} r(\mathbf{x})$$

- Interpretation : Iterative residual denoising
  - The solution is obtained by recursively subtracting from the input refined estimates of the noise realization distorting the input

$$\mathbf{x}^1 = \mathbf{y} - \varepsilon \frac{h^1(\mathbf{y})}{\max\left(\|h^1(\mathbf{y})\|_2, \varepsilon\right)} = \mathbf{x} + \left(\mathbf{n} - \mathbf{n}^1\right)$$

$$\vdots$$

$$\mathbf{x}^k = \mathbf{y} - \mathbf{n}^k = \mathbf{x} + \left(\mathbf{n} - \mathbf{n}^k\right), \quad k > 1$$

$$\mathbf{n}^k = \varepsilon \frac{\mathbf{n}^{k-1} + h^k\left(\mathbf{x}^{k-1}\right)}{\boxed{\max\left(\|\mathbf{n}^{k-1} + h^k\left(\mathbf{x}^{k-1}\right)\|_2, \varepsilon\right)}}$$

normalized noise realization estimate

Universal Network Architecture

- Cascade of composite layers (“stages”)
- Each layer implements a single PGM update

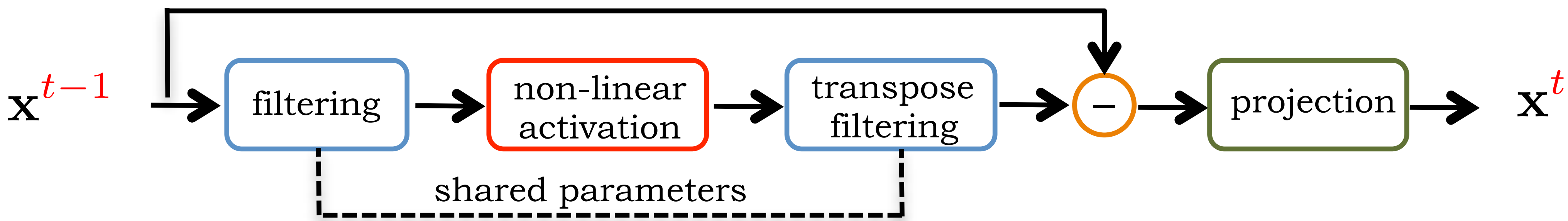
$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{x}^{t-1}\right)\right)$$

Universal Network Architecture

- Cascade of composite layers (“stages”)
- Each layer implements a single PGM update

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{x}^{t-1}\right)\right)$$

$$\Downarrow$$

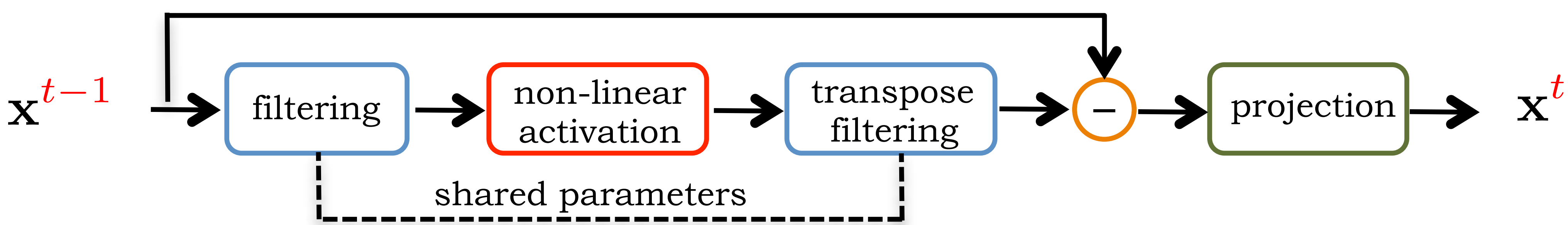


Universal Network Architecture

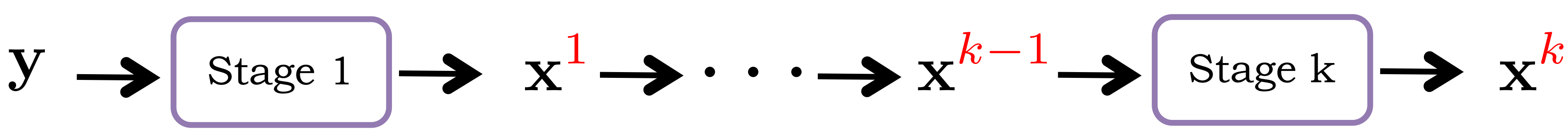
- Cascade of composite layers (“stages”)
- Each layer implements a single PGM update

$$\mathbf{x}^t = \Pi_{\mathcal{C}} \left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t \left(\mathbf{L}_k \mathbf{x}^{t-1} \right) \right)$$

$$\Downarrow$$



- Feedforward network by iteration unrolling



Lefkimmiatis CVPR `18

One-Shot Residual Denoising Network

- In MM-based schemes denoising does not need to be exhaustive
- We employ a single-iteration variant of the UDNet network

- PGM updates reminder

$$\mathbf{x}^t = \Pi_{\mathcal{C}} \left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t \left(\mathbf{L}_k \mathbf{x}^{t-1} \right) \right)$$

One-Shot Residual Denoising Network

- In MM-based schemes denoising does not need to be exhaustive
- We employ a single-iteration variant of the UDNet network

- PGM updates reminder

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{x}^{t-1}\right)\right)$$

  - For a single iteration with input the noisy image we have:

$$\mathbf{x}^{\star} = \mathbf{y} - \varepsilon \frac{\sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{y}\right)}{\max\left(\left\|\sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{y}\right)\right\|_2, \varepsilon\right)}$$

Kokkinos & Lefkimmiatis ECCV `18

One-Shot Residual Denoising Network

- In MM-based schemes denoising does not need to be exhaustive
- We employ a single-iteration variant of the UDNet network

- PGM updates reminder

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^\mathsf{T}\psi^t\left(\mathbf{L}_k\mathbf{x}^{t-1}\right)\right)$$

  - For a single iteration with input the noisy image we have:

$$\mathbf{x}^\star = \mathbf{y} - \Pi_2^{\varepsilon}\left(\sum_{k=1}^{K} \mathbf{L}_k^\mathsf{T}\psi^t\left(\mathbf{L}_k\mathbf{y}\right)\right)$$

One-Shot Residual Denoising Network

- In MM-based schemes denoising does not need to be exhaustive
- We employ a single-iteration variant of the UDNet network

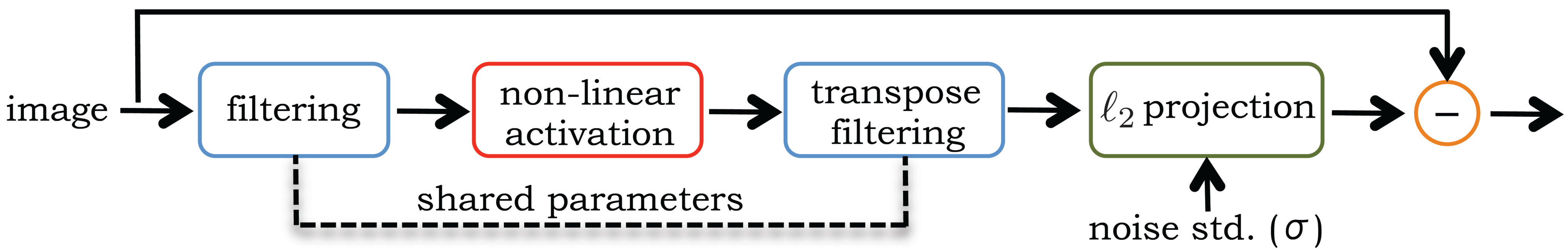
- PGM updates reminder

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{x}^{t-1}\right)\right)$$

  - For a single iteration with input the noisy image we have:

$$\mathbf{x}^{\star} = \mathbf{y} - \Pi_2^{\varepsilon}\left(\sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{y}\right)\right)$$

$$\Downarrow$$



One-Shot Residual Denoising Network

- In MM-based schemes denoising does not need to be exhaustive
- We employ a single-iteration variant of the UDNet network

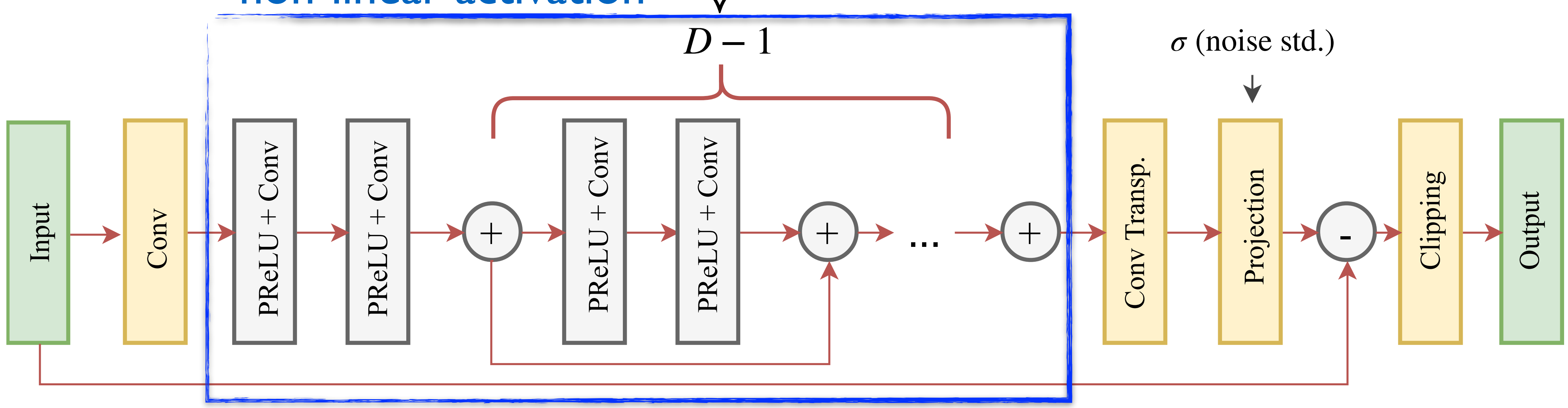
- PGM updates reminder

$$\mathbf{x}^t = \Pi_{\mathcal{C}}\left(\mathbf{x}^{t-1} - \sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{x}^{t-1}\right)\right)$$

  - For a single iteration with input the noisy image we have:

$$\mathbf{x}^\star = \mathbf{y} - \Pi_2^{\varepsilon}\left(\sum_{k=1}^{K} \mathbf{L}_k^{\mathsf{T}} \psi^t\left(\mathbf{L}_k \mathbf{y}\right)\right)$$

non-linear activation



Kokkinos & Lefkimmiatis ECCV `18

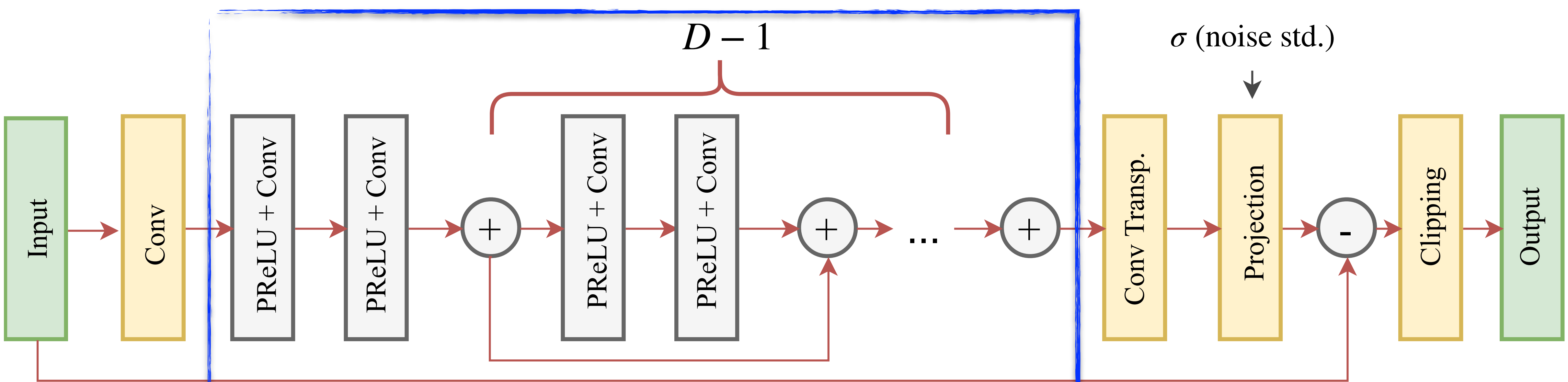
MM-based Burst Image Restoration Recap

- Recast the burst imaging problem to a series of **image denoising problems**

$$\tilde{Q}\left(\mathbf{x};\mathbf{x}^{(t)}\right)=\frac{1}{2\sigma^2}\left\|\mathbf{x}-\mathbf{z}\right\|_2^2+r\left(\mathbf{x}\right)+\text{const.}$$

$$\mathbf{z}=\mathbf{x}^{(t)}+\frac{1}{B}\sum_{i=1}^{B}\mathbf{S}_i^T\mathbf{H}^T\left(\mathbf{y}_i-\mathbf{H}\mathbf{S}_i\mathbf{x}^{(t)}\right)$$

- In each iteration a refined estimate of the solution is computed
- Instead of solving an optimization problem we employ a residual denoising network



non-linear activation

Iterative Neural Network

Algorithm 1: Proposed Iterative Neural Network for bust photography applications

Input: $\mathbf{H}$: Degradation Operator, $\mathbf{y}_{\{1\ldots B\}}$: input burst, K: iterations, $\boldsymbol{w} \in \mathbb{R}^K$: extrapolation weights, σ: estimated noise, $\boldsymbol{s} \in \mathbb{R}^K$: projection parameters

$\mathbf{x}^0 = \mathbf{0}$;

Initialize $\mathbf{x}^1$ using $\mathbf{y}_{ref}$;

Estimate mappings $\mathbf{S}_{1\ldots B}$;

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

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

  $\mathbf{z} = 0$;

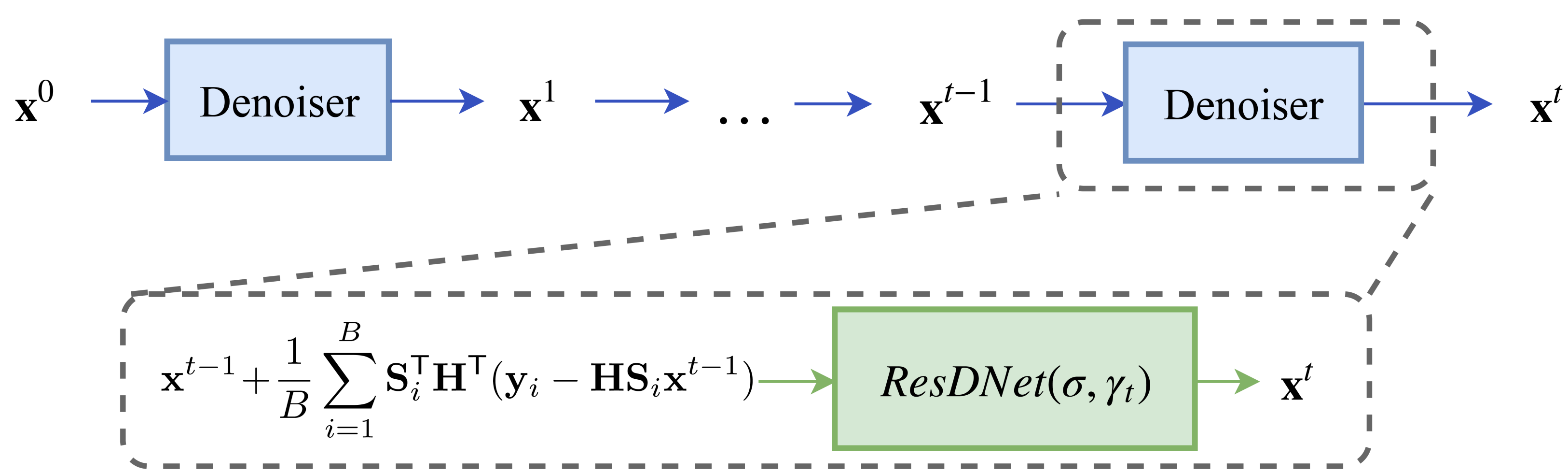
  for $i \leftarrow 1$ **to** B **do**

    $\mathbf{z} = \mathbf{z} + \mathbf{S}_i^\mathsf{T}\mathbf{H}^\mathsf{T}(-\mathbf{y}_i + \mathbf{H}\mathbf{S}_i\mathbf{u})$;

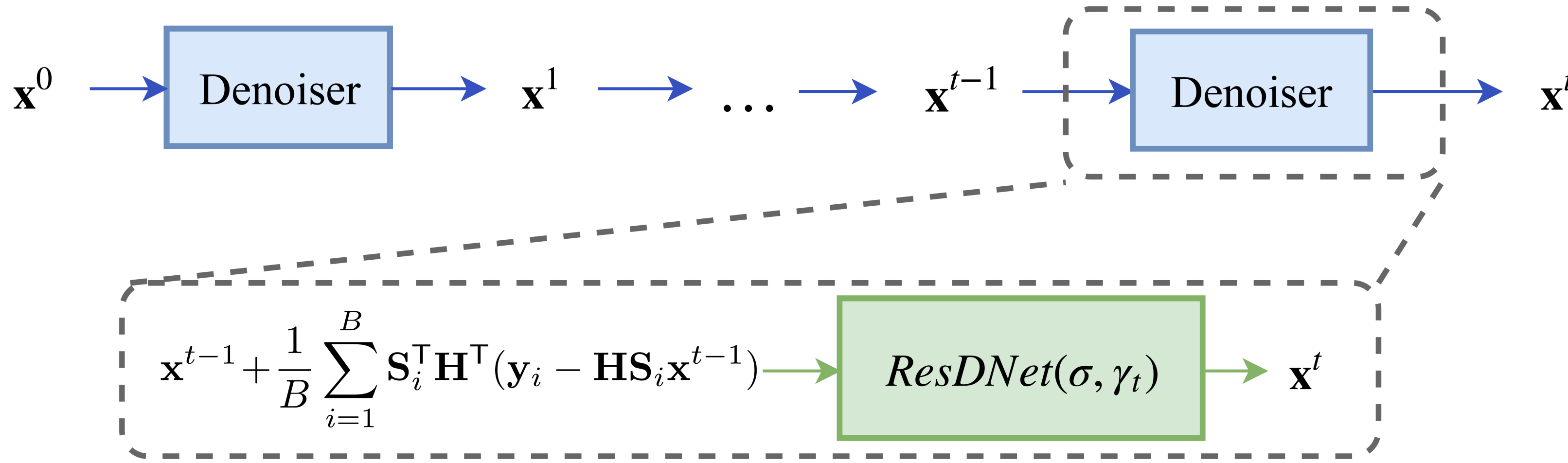
  end

  $\mathbf{x}^{t+1} = \mathrm{ProxNet}(\mathbf{x}^t - \mathbf{z}/B, \sigma, \boldsymbol{s}_t)$;

end



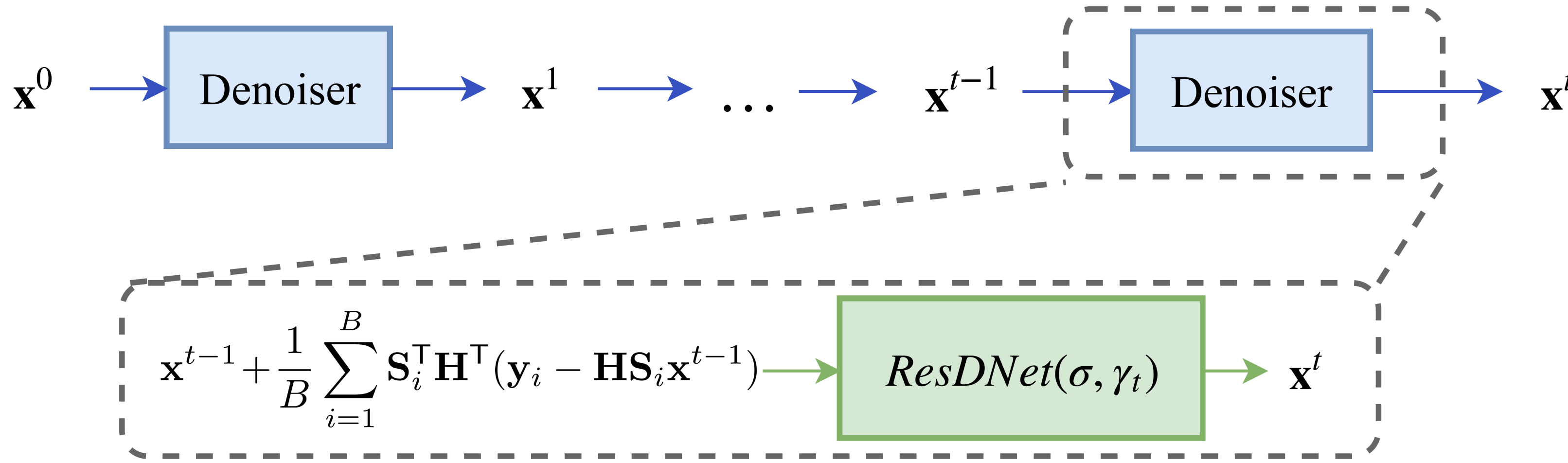
Iterative Neural Network - Training



- The network parameters are **shared** across iterations
  - ✓ small memory footprint of the network

Kokkinos & Lefkimmiatis CoRR `18

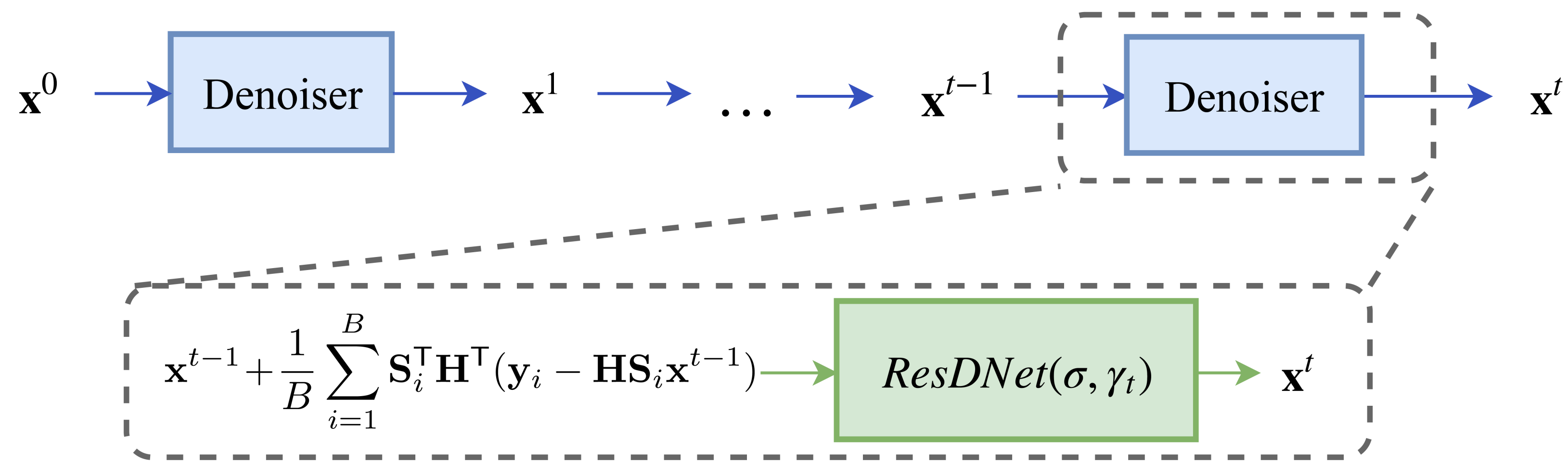
Iterative Neural Network - Training



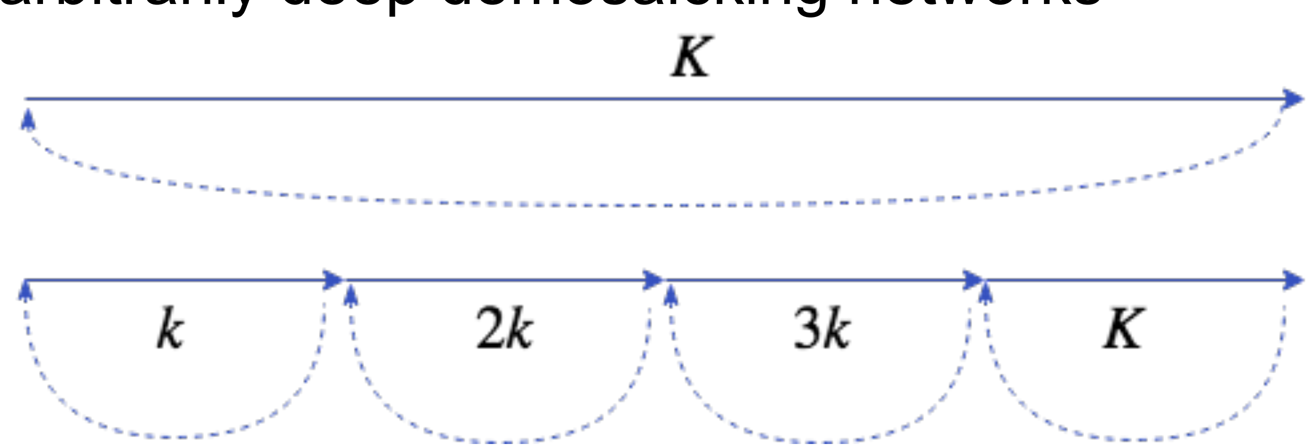
- The network parameters are **shared** across iterations
  - ✓ small memory footprint of the network
- Challenge: How to efficiently train the network by back-propagation?
  - Back propagation through time (BPTT) is **extremely memory demanding**

Kokkinos & Lefkimmiatis CoRR `18

Iterative Neural Network - Training



- The network parameters are **shared** across iterations
  - ✓ small memory footprint of the network
- Challenge: How to efficiently train the network by back-propagation?
  - Back propagation through time (BPTT) is **extremely memory demanding**
- Use instead Truncated BPTT
  - adopted approach in natural language processing (NLP)
  - first to use in image restoration or computer vision problems
  - ✓ We can train arbitrarily deep demosaicking networks



Kokkinos & Lefkimmiatis CoRR `18

Comparisons - Evaluation

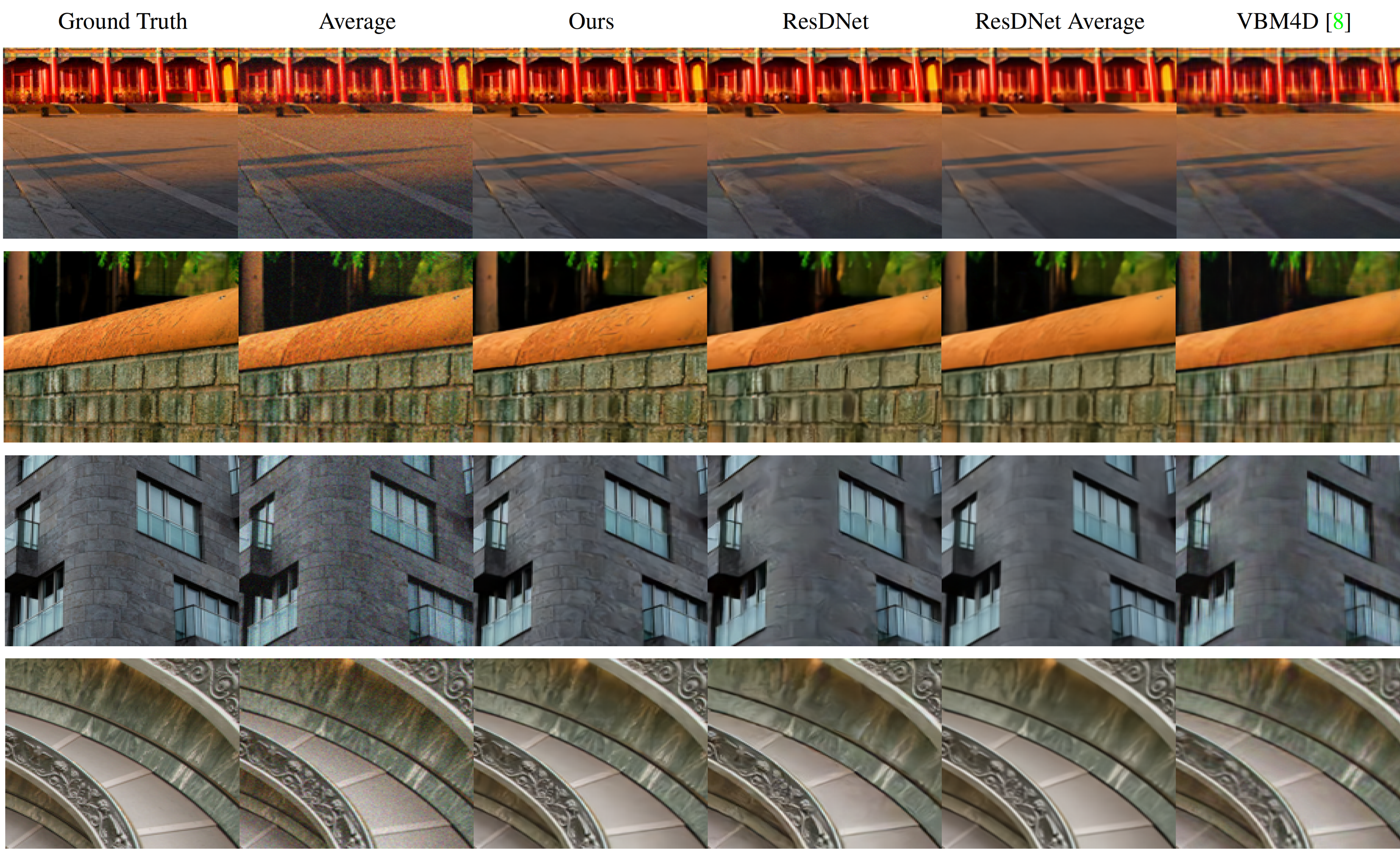
burst demosaicking (MSR dataset)

	noisy		noise-free	
	linRGB	sRGB	linRGB	sRGB
Bilinear				
- single	27.62	23.02	29.07	22.86
- burst	30.03	26.45	31.46	27.23
Gharbi [10]				
- single	36.52	31.37	41.08	34.46
- burst	37.14	31.87	39.74	34.39
Kokkinos [21]				
- single	38.48	33.41	41.03	34.37
- burst	38.06	33.06	38.93	33.02
BM3D-CFA[7]				
- single	35.63	30.49	-	-
- burst	35.36	30.30	-	-
Ours	**39.64**	**34.56**	**42.40**	**36.24**
Ours (oracle)	41.55	35.59	42.40	36.24

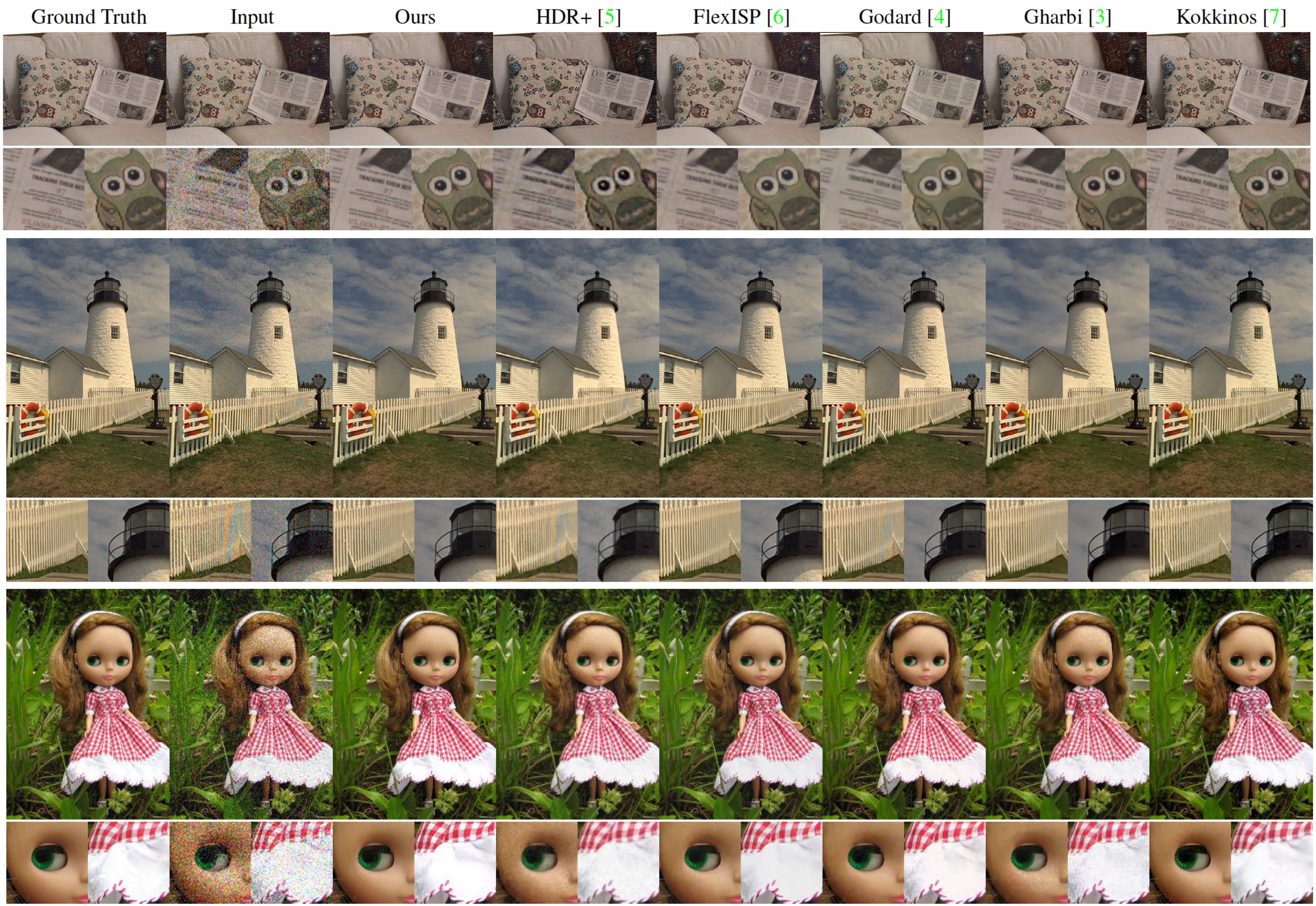
burst Gaussian denoising (Waterloo dataset)

Methods	σ=5	σ=10	σ=15	σ=20	σ=25
noisy ref. frame	34.26	28.37	24.95	22.55	20.71
BM3D	39.78	35.86	33.55	31.86	30.50
VBM4D	39.64	35.67	33.35	31.67	30.34
ResDNet:					
- single	**40.19**	36.65	34.55	33.03	31.82
- burst	39.69	37.65	36.06	34.89	33.86
Ours	40.08	**38.71**	**37.36**	**36.24**	**35.28**

Burst Gaussian Denoising (σ = 25)

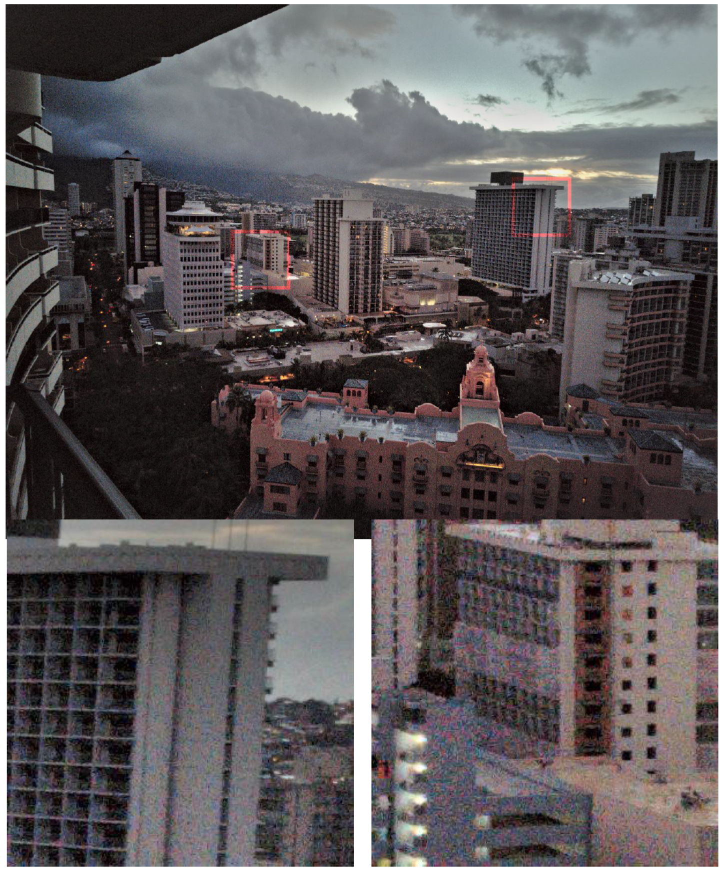


Burst Demosaicking (FlexISP Dataset)

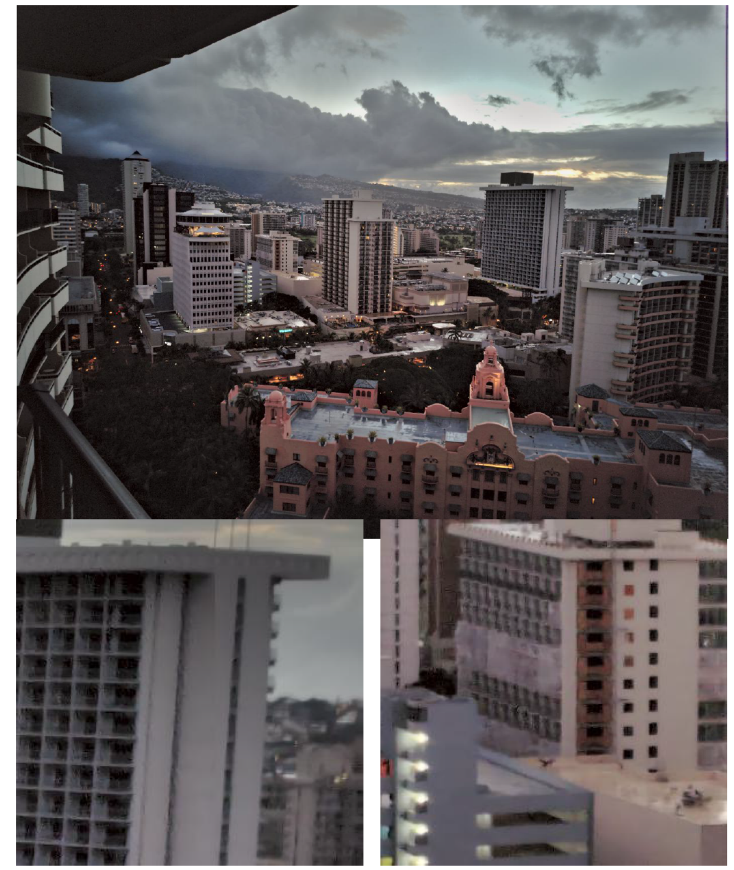


Burst Denoising-Demosaicking (real data)

Input

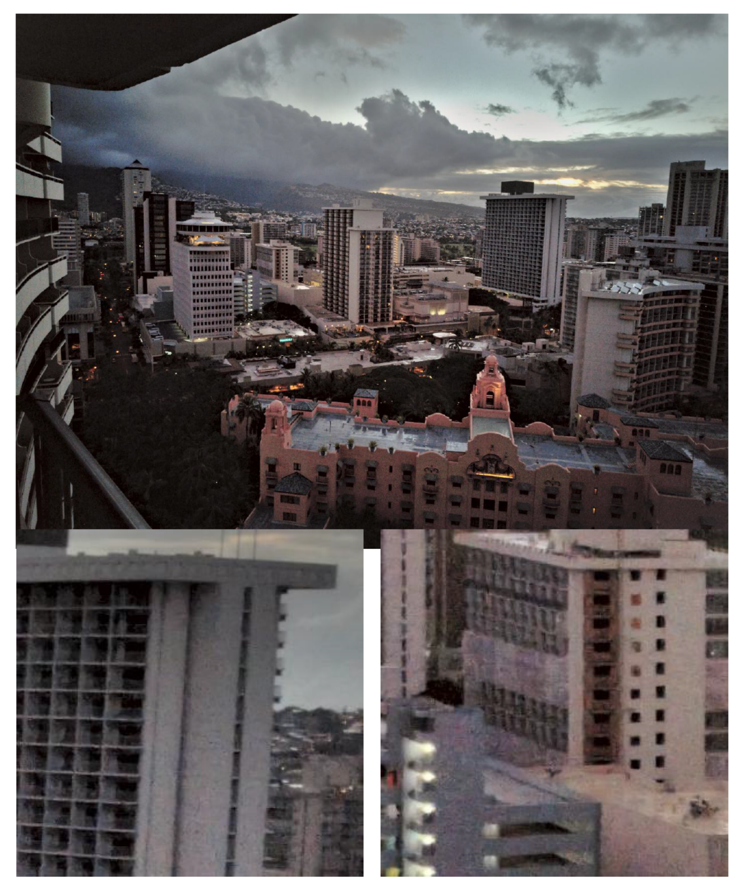


Ours



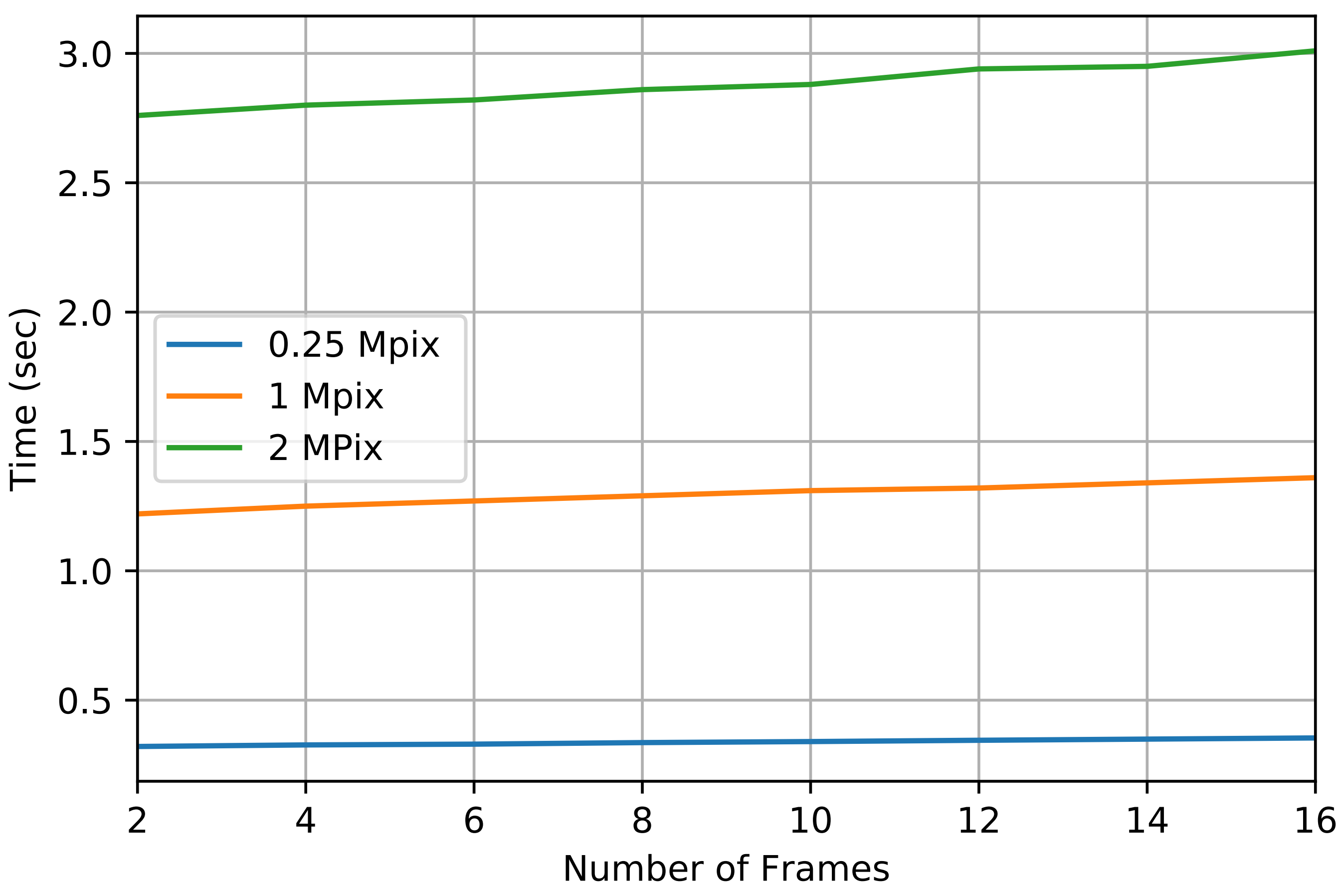
HDR+

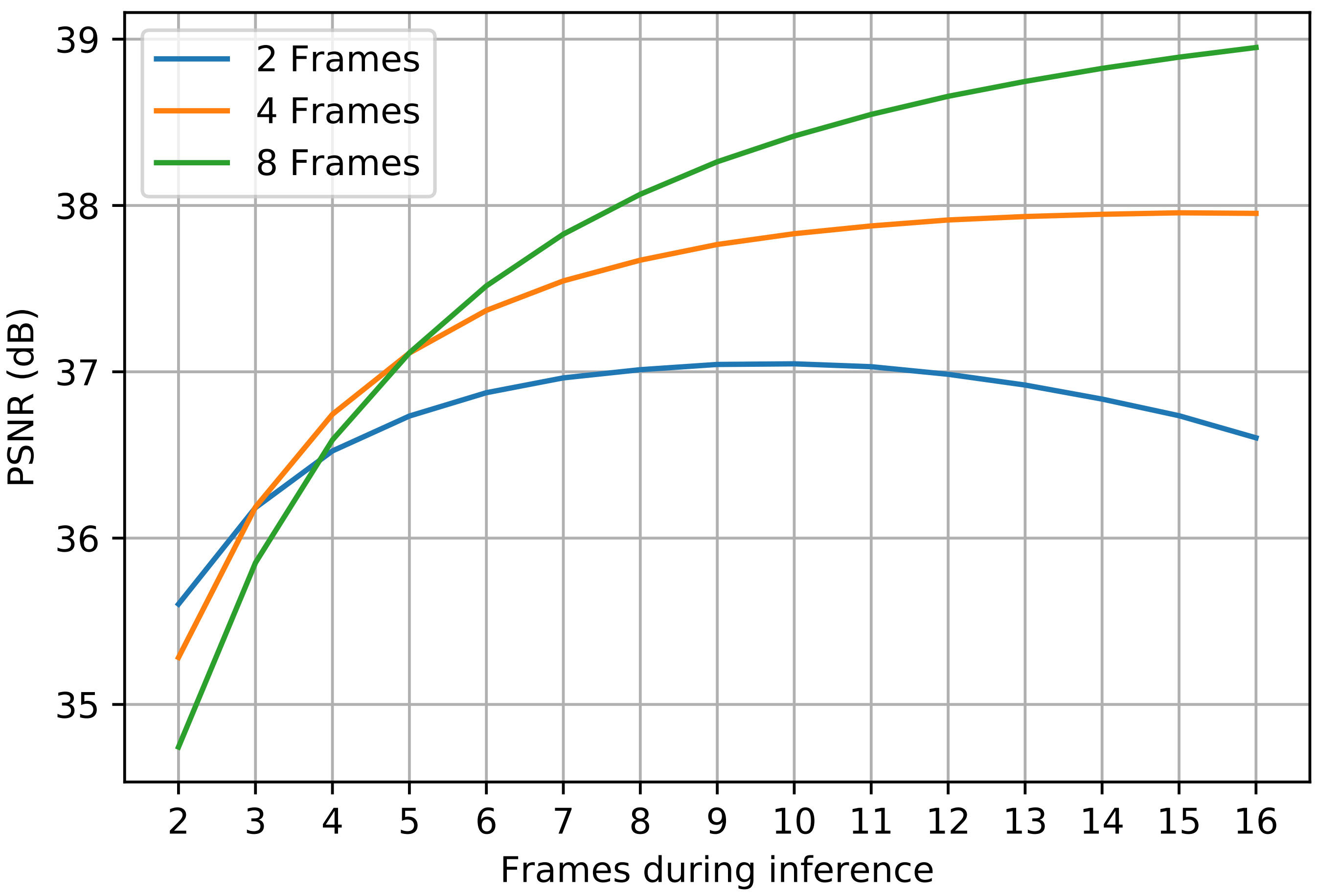
Kokkinos



Network Properties

- The proposed network architecture exhibits some important properties:
  - ✓ invariant to the ordering of the burst frames
  - ✓ good generalization ability to a different number of frames than those used for training
  - ✓ Minimal computational overhead if we increase the burst size





Summary and Future Research Directions

- **Deep Learning for inverse imaging problems**
  - Principled network design motivated by variational and optimization methods
  - Data-driven approach for learning the regularization operator
  - Single set of network parameters that handles a wide range of noise levels
  - Computational efficiency and increased robustness to degradation factors
  - Good generalization ability of the developed networks to real noise
  - **State-of-the-art** image restoration results using considerably smaller networks than previous approaches
  - Introduction of a generic framework that utilizes the proposed denoising networks as sub-solvers for dealing with other general inverse problems
  - Efficient training of iterative networks through Truncated BBTT

References and Main Collaborators

- S. Lefkimmiatis, “Non-Local Color Image Denoising with Convolutional Neural Networks”, IEEE Int. Conference on Computer Vision and Pattern Recognition, 2017
- S. Lefkimmiatis, “Universal Denoising Networks : A Novel CNN Architecture for Image Denoising, IEEE Int. Conference on Computer Vision and Pattern Recognition, 2018
- F. Kokkinos and S. Lefkimmiatis, “Deep Image Demosaicking using a Cascade of Convolutional Residual Denoising Networks”, European Conference on Computer Vision, 2018
- F. Kokkinos and S. Lefkimmiatis, “Iterative Residual Network for Deep Joint Image Demosaicking and Denoising” , CoRR:1807.06403
- F. Kokkinos and S. Lefkimmiatis, “Iterative Residual CNNs for Burst Photography Applications”, soon on Arxiv

Filippos Kokkinos
(Skoltech)

References and Main Collaborators

- S. Lefkimmiatis, “Non-Local Color Image Denoising with Convolutional Neural Networks”, IEEE Int. Conference on Computer Vision and Pattern Recognition, 2017
- S. Lefkimmiatis, “Universal Denoising Networks : A Novel CNN Architecture for Image Denoising, IEEE Int. Conference on Computer Vision and Pattern Recognition, 2018
- F. Kokkinos and S. Lefkimmiatis, “Deep Image Demosaicking using a Cascade of Convolutional Residual Denoising Networks”, European Conference on Computer Vision, 2018
- F. Kokkinos and S. Lefkimmiatis, “Iterative Residual Network for Deep Joint Image Demosaicking and Denoising” , CoRR:1807.06403
- F. Kokkinos and S. Lefkimmiatis, “Iterative Residual CNNs for Burst Photography Applications”, soon on Arxiv

Filippos Kokkinos
(Skoltech)

Thank you for your attention

software and publications available at: http://cig.skoltech.ru