RBSRICNN: Raw Burst Super-Resolution through Iterative Convolutional Neural Network

Rao Muhammad Umer, Christian Micheloni

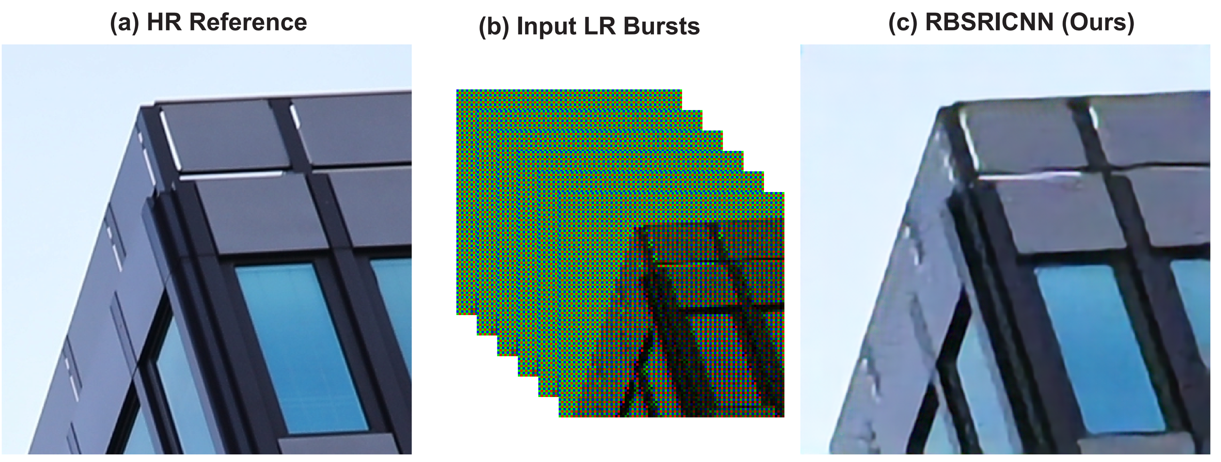
University of Udine, Italy.

Code: https://git.io/JXw0T

Problem Definition and Motivations

Goal:

- The Burst Super-resolution is the task of *fusing several low-resolution (LR) frames* to produce a *single high-resolution (HR)* image.



Motivations:

- Compared to *DSLR cameras*, LR images are usually obtained in many portable *mobile devices* with *compact camera sensors* due to their *physical limitations*.
- Due to the *ill-posed* nature of the SISR problem, the existing SR methods have limited performance to recover high frequency details through *single image learned priors*.
- On the other hand, the *Multi-Frame Super-Resolution (MFSR)* aims to recover the latent HR image using *multiple LR frames* by exploiting the additional signal information due to *sub-pixel shifts*.
- Moreover, the existing Burst SR methods are *black-box data-driven* approaches with *larger model size* due to *not* directly model the *image formation process*.

Problem Formulation

Image forward observation model:

$$\mathbf{y}_i = \mathbf{MHS}_i(\tilde{\mathbf{x}}) + \eta_i, \quad i = 1, \ldots, B \qquad (1)$$

where, $\mathbf{y}_i$ is the *i-th* observed image of the LR burst B images, $\mathbf{M}$ is a *mosaicking operator* (*i.e.*, usually Bayer CFA), $\mathbf{H}$ is a *down-sampling operator* (*i.e.*, bilinear, bicubic, etc.), $\mathbf{S}_i$ is an *affine transformation* of the coordinate system of the image $\tilde{\mathbf{x}}$ (*i.e.*translation and rotation), and η_i is an additive *heteroskedastic Gaussian noise* related to photon shot and read noise.

Objective Function Minimization Strategy:

- We want to recover the underlying image $\mathbf{x}$ as the minimizer of the objective function:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2 B} \sum_{i=1}^{B} \|\mathbf{y}_i - \mathbf{MHS}_i(\mathbf{x})\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \qquad (2)$$

- The Eq. (2) can be also written as:

$$\mathbf{J}(\mathbf{x}) = \arg\min_{\mathbf{x}} \frac{1}{2\sigma^2 B} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \mathcal{R}(\mathbf{x}), \qquad (3)$$

where, $\mathbf{A}$=$\mathbf{MHS}$ corresponds to the *camera response*.

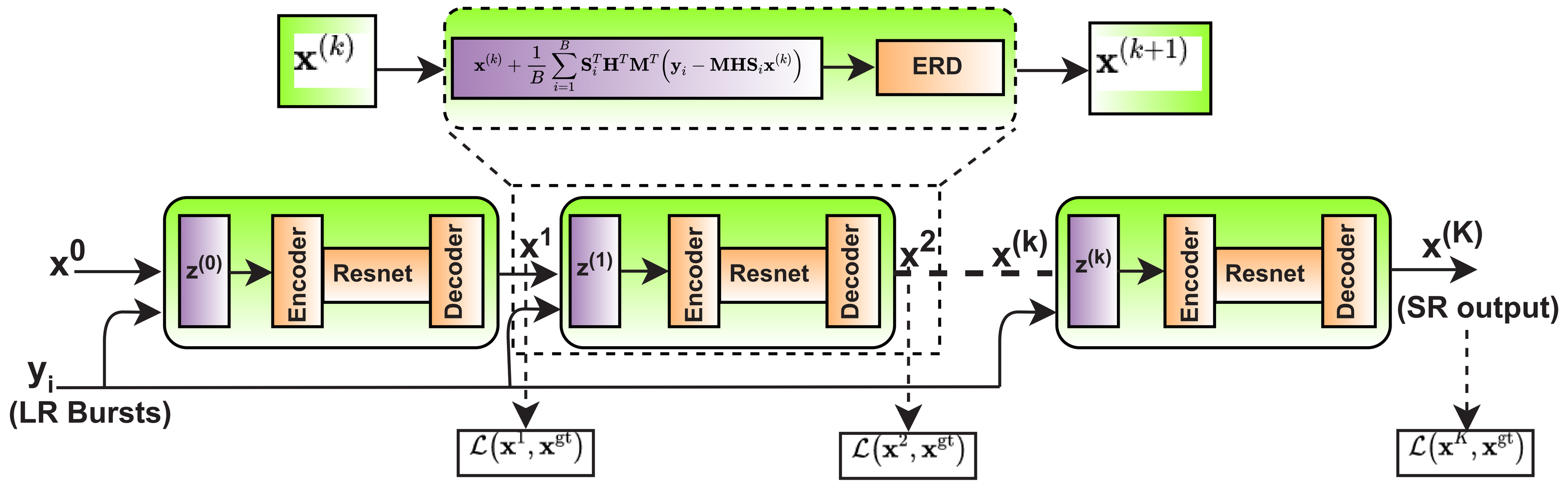
- By using the *Majorization-Minimization* framework, we have final form of the solution:

$$\begin{aligned} \hat{\mathbf{x}}^{(k)} &= \arg\min_{\mathbf{x}} \mathbf{Q}(\mathbf{x}; \mathbf{x}^{(k)}) \\ &= \tilde{d}\left(\mathbf{x}; \mathbf{x}^{(k)}\right) + \lambda \mathcal{R}(\mathbf{x}) \\ &= \frac{\alpha}{2\sigma^2 B} \|\mathbf{x} - \mathbf{z}^k\|_2^2 + \lambda \mathcal{R}(\mathbf{x}) + const. \\ &= \mathrm{Prox}_{(\lambda/\alpha\sigma^2)\mathcal{R}(.)}(\mathbf{z}^k) \end{aligned} \qquad (4)$$

where, $\mathbf{z}^k$=$\mathbf{x}^k$+$\mathbf{A}^T(\mathbf{y}-\mathbf{A}\mathbf{x}^k) \Rightarrow$
$\mathbf{z}^k$=$\mathbf{x}^{(k)}$+$\frac{1}{B}\sum_{i=1}^{B} \mathbf{S}_i^T \mathbf{H}^T \mathbf{M}^T \left(\mathbf{y}_i - \mathbf{MHS}_i \mathbf{x}^{(k)}\right)$ (See the Network Architecture diagram).

Network Architecture and Training

We *unroll* the RBSRICNN into K stages, where each stage computes the *refined estimate* of the SR image.



Loss function for the network training: We use the following function to minimize the ℓ_1-*Loss* between the estimated latent SR image ($\mathbf{x}^{(k)}$) and ground-truth (GT) ($\mathbf{x}^{(gt)}$) after k-steps as:

$$\mathcal{L} = \arg\min_{\Theta} \mathcal{L}(\mathbf{\Theta}) = \frac{1}{2} \sum_{i=1}^{N} \|\mathbf{x}_i^k - \mathbf{x}_i^{gt}\|_1 \qquad (5)$$

Experiments & Results

Dataset:

- **Synthetic Burst SR data:** Use the $46,839$ and 1204 sRGB images from the **Zurich RAW to RGB** dataset for the training and the validation, respectively. The sRGB image is first converted to the *Raw (linear) sensor* space using an *inverse camera pipeline*, then the LR burst is generated by applying *random translations and rotations*, followed by *bilinear downsampling*, further *mosaicked* and corrupted by *random noise*.
- **Real Burst SR data:** Contains testset of 639 real-world LR bursts, where each burst sequence contains 14 RAW images captured using a handheld smartphone camera using *identical camera settings* (*e.g.*, exposure, ISO) resulting in a small *random offset* between the images within the burst.

Quantitative Results:

Comparison with other Burst SR methods on $\times 4$ upscaling factor:

Burst SR Method	#Params [M]	#Conv2d	Synthetic data			Real data			Fine-tuned on Real data
			PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	
DeepJoint + RRDB	17.26	371	33.25	0.881	0.195	42.13	0.957	0.088	✓
DeepBurstSR	5.25	48	34.48	**0.905**	**0.118**	**45.17**	**0.978**	**0.037**	✓
HighRes-net	1.11	25	34.30	0.891	0.170	43.99	0.972	0.051	✓
RBSRICNN (ours)	**0.38**	**12**	**37.62**	0.895	0.166	41.40	0.952	0.101	✗

Impact of different number of input burst frames (B) and number of iterative steps (K):

Burst Size (B)	iterative steps ($K=5$)			iterative steps ($K=10$)		
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
2	34.19	0.8790	0.2498	34.12	0.8777	0.2480
4	34.69	0.8852	0.2359	34.66	0.8842	0.2317
8	35.09	0.8887	0.2277	34.99	0.8876	0.2217
14	35.12	0.8896	0.2255	35.30	0.8903	0.2165
16	**35.21**	**0.8907**	**0.2232**	35.30	**0.8909**	0.2168
32	35.23	0.8902	0.2236	**35.41**	**0.8909**	**0.2159**

Visual Results: