Iterative Residual CNNs for Burst Photography Applications

Filippos Kokkinos Stamatios Lefkimmiatis
Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia
filippos.kokkinos@skoltech.ru s.lefkimmiatis@skoltech.ru

Abstract

Modern inexpensive imaging sensors suffer from inherent hardware constraints which often result in captured images of poor quality. Among the most common ways to deal with such limitations is to rely on burst photography, which nowadays acts as the backbone of all modern smartphone imaging applications. In this work, we focus on the fact that every frame of a burst sequence can be accurately described by a forward (physical) model. This, in turn, allows us to restore a single image of higher quality from a sequence of low-quality images as the solution of an optimization problem. Inspired by an extension of the gradient descent method that can handle non-smooth functions, namely the proximal gradient descent, and modern deep learning techniques, we propose a convolutional iterative network with a transparent architecture. Our network uses a burst of low-quality image frames and is able to produce an output of higher image quality recovering fine details which are not distinguishable in any of the original burst frames. We focus both on the burst photography pipeline as a whole, i.e., burst demosaicking and denoising, as well as on the traditional Gaussian denoising task. The developed method demonstrates consistent state-of-the art performance across the two tasks and as opposed to other recent deep learning approaches does not have any inherent restrictions either to the number of frames or their ordering.

1. Introduction

With more than one billion smartphones sold each year, smartphone cameras have dominated the photography market. However, to allow for small and versatile sensors, inevitably manufacturers of such cameras need to make several compromises. As a result, the quality of images captured by smartphone cameras is significantly inferior compared to the quality of images acquired by sophisticated hand-held cameras like DSLRs. The most common hardware restrictions in smartphone cameras are the lack of large aperture lens and the small sensors that consist of fewer photodiodes. To overcome such inherent hardware restrictions, the focus is thus shifted towards the software of the camera, i.e., the Image Processing Pipeline (ISP).

The shortcomings of mobile photography can be mitigated by the use of burst photography, where a camera firstly captures a burst of images, milliseconds apart, and afterward fuses them in a sophisticated manner to produce a higher-quality image. Therefore, burst photography allows inexpensive hardware to overcome mechanical and physical constraints and thus achieving higher imaging quality in the expense of computation time. While ideally, we would like each frame of the burst to capture precisely the same scene, this is not possible due to camera motion (e.g. hand shake), scene motion by dynamic moving objects and finally inefficiencies of Optical Image Stabilization (OIS) hardware that may cause a slight drift even for completely static scenes. Therefore, homography estimation and alignment usually is necessary when processing frames of the same scene.

The idea of using a sequence of photographs to enhance the image quality, is not new and it has been successfully exploited in the past for the tasks of image debluring [1, 5], denoising [29] and super-resolution [9]. Inspired from these works, we design a restoration algorithm that involves a neural network, to handle various tasks of burst photography. First, we rely on a physical model for the observations of the burst, which in turn enables us to derive an optimization scheme for restoration purposes. The optimization scheme is combined with supervised learning of a neural network with a transparent architecture, leading to an Iterative Neural Network (INN). The developed framework exhibits by design many desired properties, which competing deep learning methods for burst photography do not necessarily exhibit, namely a) inherent invariance to the ordering of the frames, b) support of bursts of arbitrary size and c) scalability to burst size.

2. Related work

2.1. Image Denoising

Single image denoising is a longstanding problem, and it has progressed dramatically in recent decades, approaching its believed performance limit [26]. The list of methods includes but not limited to Field-of-Experts [35], Non-Local Means [4] and BM3D [6], with the latter being the de

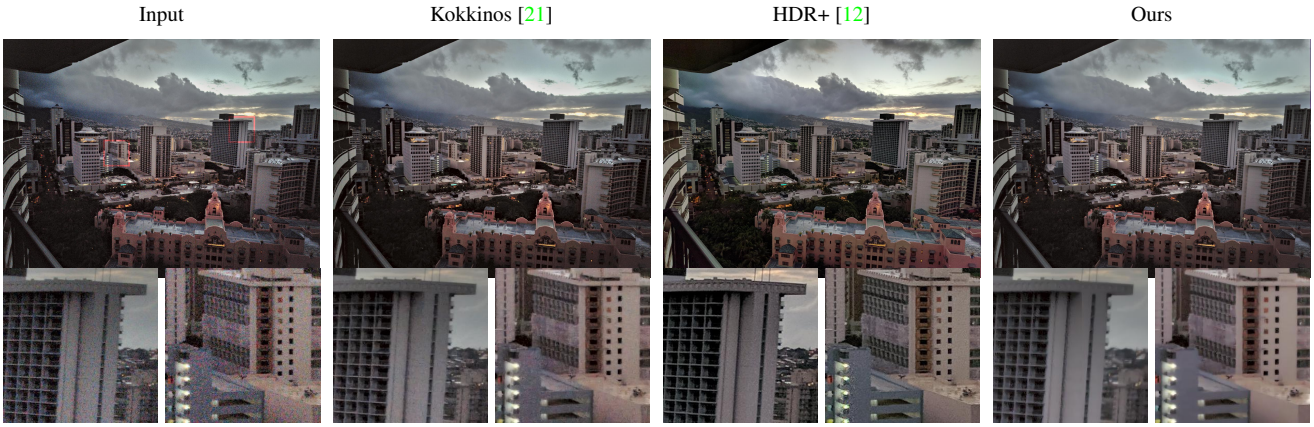

Figure 1: Demosaicking and denoising of a real low-light raw burst from the HDR+ dataset [12]. Our method achieves high quality reconstruction even in cases of excessive noise in the sensor data.

facto method used till today. With the advent of deep learning, several learning-based methods have emerged during the last few years that take advantage of neural networks in order to push the reconstruction quality even further. Systems like DnCNN [41], NLNet [23] and MemNet [37] have succeeded to set a new state-of-the art performance for the image denoising task. Unfortunately, recent works empirically indicate that we are now close to the believed performance limit for single image denoising task, since quantitative performance improvements are no longer substantial and do not fully justify the simultaneous disproportionate increase of computational complexity.

Fortunately, burst denoising still allows the development of methods that can achieve better reconstruction than single image denoising. In fact, several multi-frame variants of single-image denoising methods have been successfully developed. For example, VBM3D [22] and VBM4D [29] are two known extensions of the BM3D framework that work on videos and bursts of images, respectively. Furthermore, techniques as in [43] were developed specifically for low resource photography applications and denoise an image using a burst sequence, in a fraction of the time required by VBM4D and other variants. Finally, modern deep learning approaches for burst denoising have recently emerged, such as those in [11, 31], and provide insights for the success of end-to-end methods by achieving superior reconstruction quality.

2.2. Image Demosaicking

While the literature on multi-image demosaicking methods falls short, demosaicking as a standalone problem has been studied for decades and for a complete survey we refer to [40]. A very common approach is bilinear interpolation, as well as, other variants of this method which are adaptive to image edges [18, 30]. During the last years, the image demosaicking task witnessed an incredible quantitative and qualitative performance increase via the use of neural network approaches like those in [10, 17] and most recently in [21]. This performance increase holds true even under the presence of noise perturbing the camera sensor readings.

Related to multi-frame photography, two well known systems that support burst demosaicking are FlexISP [16] and ProxImaL [15], which offer end-to-end formulations and joint solution via efficient optimization for many image processing related problems. Finally, a very successful commercial application on burst photography reconstruction is HDR+, introduced in [12], where a burst of frames is utilized to alleviate shortcomings of smartphone cameras such as low dynamic range and noise perturbations.

3. Problem formulation

To solve a variety of burst photography problems, we rely on the following observation model for each frame $\mathbf{y}_i$ of a burst sequence of total size B,

$$\mathbf{y}_i = \mathbf{H} S_i(\mathbf{x}) + \mathbf{n}_i,\ i = 1 \ldots, B. \tag{1}$$

In Eq. (1), $\mathbf{y}_i \in \mathbb{R}^N$ corresponds to the degraded version of the affinely transformed underlying image $\mathbf{x} \in \mathbb{R}^N$, which we aim to restore. While $\mathbf{x}$ and $\mathbf{y}_i$ are two dimensional images, for the sake of mathematical derivations, we assume that they have been raster scanned using a lexicographical order, and they correspond to vectors of N dimensions. The operator $S_i : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is responsible for the affine transformation of the coordinate system of $\mathbf{x}$. Specifically, it provides a mapping by interpolating values for each frame i from the grid of the original image $\mathbf{x}$. In our proposed method, we restrict the affine transformations to be rotational and translation so as to be on par with realistic burst photography applications. While in the above model it is

assumed that the affine transformation is known, in practice we can only estimate it from the observations $\mathbf{y}_i$ by setting one observation as a reference and aligning all other observations to the reference. This reference frame is considered to be completely aligned to the underlying image $\mathbf{x}$, and their relationship is described as $\mathbf{y}_{ref} = \mathbf{H}\mathbf{x} + \mathbf{n}_{ref}$. Additionally, the underlying image $\mathbf{x}$ is further distorted by a linear operator $\mathbf{H} \in \mathbb{R}^{N\times N}$, which describes a specific restoration problem that we aim to solve. This formulation is one of the most frequently used in the literature to model a variety of restoration problems such as image inpainting, deconvolution, demosaicking, denoising, and super-resolution. Each observation $\mathbf{y}_i$ is also distorted by noise $\mathbf{n}_i \sim \mathcal{N}(0,\, \sigma^2)$, which is assumed to follow an i.i.d Gaussian distribution.

Recovering $\mathbf{x}$ from the measurements $\mathbf{y}_i$ belongs to the broad class of inverse problems. For most practical problems, the operator $\mathbf{H}$ is typically singular, i.e., not invertible. This fact, coupled with the presence of noise perturbing the measurements and the affine transformation leads to an ill-posed problem where a unique solution does not exist. In general, such problems can be addressed following a variational approach. Under this framework, a solution is obtained by minimizing an objective function of the form:

$$\mathbf{x}^\star = \underbrace{\arg\min_{\mathbf{x}} \frac{1}{2\sigma^2 B} \sum_{i=1}^{B} \|\mathbf{y}_i - \mathbf{H}S_i(\mathbf{x})\|_2^2}_{f(\mathbf{x})} + r(\mathbf{x}), \quad (2)$$

where the first term corresponds to the data fidelity term that quantifies the proximity of the solution to the observations and the second term corresponds to the regularizer of the solution, which encodes any available prior knowledge we might have about the underlying image. As it can be seen from Eq. (2), the solution $\mathbf{x}^\star$ must obey the observation model for each frame $\mathbf{y}_i$ of the burst. While the above variational formulation is general enough to accommodate for a variety of different inverse problems, in Section 7 we focus on two particular problems: 1) joint demosaicking and denoising and 2) burst Gaussian denoising. In the first case, $\mathbf{H}$ becomes a binary diagonal matrix that corresponds to the Color Filter Array (CFA) of the camera, while in the second case $\mathbf{H}$ reduces to the identity operator.

As we mentioned earlier, the role of the regularizer is to promote solutions that follow specific image properties and as a result its choice significantly affects the end-result of the restoration. Some typical choices for regularizing inverse problems is the Total Variation [36] and the Tikhonov [38] functionals. While such regularizers have been frequently used in the past in image processing and computer vision applications, their efficacy is limited. For this reason, in this work, we follow a different path, and we attempt to learn the regularizer implicitly from available training data. Therefore, throughout this work, we do not make any assumptions regarding the explicit form of the regularizer. Rather, as we will explain later in detail, our goal is to learn the effect of the regularizer to the solution through the proximal map [32].

4. Proximal gradient descent

Efficiently solving Eq. (2) has been a longstanding problem, and as a result a variety of sophisticated optimization methods have been proposed over the years. In our work, we employ a relatively simple method that extends the classical gradient descent, namely the Proximal Gradient Descent (PGD) [32]. In particular, PGD is a generalization of gradient descent that can deal with the optimization of functions that are not fully differentiable but they can be split into a differentiable and a non-differentiable part, i.e. $F(\mathbf{x}) = s(\mathbf{x}) + g(\mathbf{x})$. Then, according to PGD, the solution can be obtained in an iterative fashion as follows:

$$\mathbf{x}^t = \mathrm{prox}_{\gamma g}(\mathbf{x}^{t-1} - \gamma \nabla_{\mathbf{x}} s(\mathbf{x}^{t-1})), \quad (3)$$

where γ is the step size and $\mathrm{prox}_{\gamma g}$ is the proximal operator, related to the non-smooth part of the overall function, $g(\mathbf{x})$, and the step size γ. Typically, γ is adaptive and is computed using a line-search algorithm. However, when $s(\cdot)$ is Lipschitz continuous it can be fixed and set as $\gamma = \frac{1}{L}$, where L is the Lipschitz constant of $\nabla_{\mathbf{x}} s$. In each iteration t, first a gradient descent step is performed for the smooth part $s(\mathbf{x})$ of the objective function, while in the sequel the non-smooth term is handled via the proximal operator, whose action on a vector $\mathbf{v}$ is defined as:

$$\mathrm{prox}_{\gamma g}(\mathbf{v}) = \arg\min_{\mathbf{z}} \tfrac{1}{2}\|\mathbf{v} - \mathbf{z}\|_2^2 + \gamma g(\mathbf{z}). \quad (4)$$

From a signal processing perspective, the proximal map corresponds to the regularized solution of a Gaussian denoising problem, where $\mathbf{v}$ is the noisy observation, $g(\cdot)$ is the employed regularizer and γ the regularization parameter. Based on the above and by inspecting Eq. (2), we observe that in our case the data fidelity corresponds to the smooth part while we further consider the regularizer as the non-smooth part. We note, the most effective regularizers in variational methods have been shown to be indeed non-differentiable and, thus, our assumption is a reasonable one.

Referring to Eq. (2), the gradient of the data fidelity term can be easily computed as

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{\sigma^2 B} \sum_{i=1}^{B} \nabla_{\mathbf{x}} S_i(\mathbf{x}) \mathbf{H}^{\mathsf{T}} (-\mathbf{y}_i + \mathbf{H}S_i(\mathbf{x})). \quad (5)$$

A useful observation is that the gradient of $f(\mathbf{x})$ can be linearized and therefore the time-consuming calculation of the Jacobian of the affine transform S_i can be entirely avoided. The base of this observation is that the mapping $S_i(\mathbf{x})$ corresponds to an interpolation, such as bilinear, on an image

$\mathbf{x}$ with respect to a certain warping matrix. By calculating beforehand the new pixel locations, using the estimated warping matrix, that we would like to interpolate from the image $\mathbf{x}$, the interpolation itself can be re-written as a linear operation $\mathbf{S}_i\mathbf{x}$. In this case, $\mathbf{S}_i$ is a sparse matrix with only a few of its columns being non-zero and which hold the coefficients for the weighted averaging of pixel intensities. Therefore, under this approach it holds that $\mathbf{S}_i\mathbf{x} = S_i(\mathbf{x})$. For example, in the case of bilinear interpolation only four elements of each row of the matrix $\mathbf{S}_i$ will be non-zero, while in the case of nearest neighbor interpolation only one element is non-zero and is equal to one.

Consequently, the gradient of the data fidelity term can be rewritten as

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \frac{1}{\sigma^2 B}\sum_{i=1}^{B} \mathbf{S}_i^{\mathsf{T}}\mathbf{H}^{\mathsf{T}}(-\mathbf{y}_i + \mathbf{H}\mathbf{S}_i\mathbf{x}), \qquad (6)$$

where $\mathbf{S}_i^{\mathsf{T}}$ is the adjoint operator of $\mathbf{S}_i$. This adjoint operation amounts to interpolating an image $\mathbf{x}$ with the inverse of the warping matrix. In our case, this matrix is always existent, since we have restricted our affine transformations to support only rotation and translation. Finally, by using the gradient of the data fidelity term of Eq. (6), in the proximal gradient step described in Eq. (3), and by computing its Lipschitz constant as $L = \frac{1}{\sigma^2}$ (the proof is provided in the supplementary material), we end up with the following iterative optimization step for burst photography applications

$$\mathbf{x}^t = \text{prox}_{\sigma^2 r}(\mathbf{x}^{t-1} + \frac{1}{B}\sum_{i=1}^{B} \mathbf{S}_i^{\mathsf{T}}\mathbf{H}^{\mathsf{T}}(\mathbf{y}_i - \mathbf{H}\mathbf{S}_i\mathbf{x}^{t-1})). \quad (7)$$

In order to retrieve the solution of the minimization problem in Eq. (2) based on the above iterative scheme, the appropriate form of $r(\mathbf{x})$ must be first specified. However, this is far from a detrimental task. Apart from this, the convergence to a solution usually requires a large number of iterations, which implies a significant computational cost.

To deal with these challenges, in this work we pursue a different approach than conventional regularization methods. In particular, instead of selecting a specific regularizer and deriving the solution via Eq. (7), we design a network to learn the mapping between the proximal input and the denoised output. This strategy allows us to unroll K iterations of the PGD method and use a suitable network to approximate the output of the proximal operator. It is important to note that this approach does not carry any risk of leading to a reconstruction of inferior quality. The reason is that in large scale optimization techniques, even when the regularizer is fully specified, typically the proximal map cannot be computed in closed-form. In such cases [2, 25], it is roughly approximated via iterative schemes without this jeopardizing the overall reconstruction quality.Another important point we would like to highlight is that our approach, as opposed to other related methods that use a network to replace the proximal operator such as IRCNN [42], Plug and Play [39] and RED [34], is completely parameter-free and, thus, no manual tuning is required so that a good reconstruction is produced.

5. Proposed Iterative Neural Network (INN)

5.1. Proximal Network

As described in Section 4, the proximal map can be interpreted as the regularized solution of a Gaussian denoising problem. Based on this observation, we can exploit the capabilities of neural networks and replace the iterative computation of the proximal map with a CNN that takes as input a noisy image and the standard deviation of the noise and returns as output the denoised version of the input.

While there are many image denoising neural networks such as the DnCNN [41] or MemNet[37] that we could use to approximate the proximal map, in this work we employ the ResDNet network described in [20], which was originally inspired by UDNet [24]. Similarly to DnCNN, ResDNet is a fully convolutional denoising network and can handle a wide range of noise levels by using a single set of parameters. It also has a residual architecture since instead of estimating directly the denoised image, it first estimates a noise realization which is then subtracted from the noisy input. The advantage of ResDNet over DnCNN is that it takes as an additional input the standard deviation of the noise, which is then used by the network to normalize the noise estimate in order to ensure that it has the desired variance. This feature is instrumental for the successful implementation of our overall scheme, as it allows us to have more control over the output of the network.

In detail, the architecture of ResDNet consists of D residual blocks with 2 convolutional layers each of 64 filters and kernels with dimensionality 3×3. The residual blocks precedes a convolutional layer applied on the input which increases the number of channels from 3 to 64 using kernels of size 5×5. The feature maps are eventually decreased to 3 from 64 via a convolutional layer with a kernel of support 5×5. In every step, the employed non-linearity, which is applied after every convolutional layer, except the last one, is the parametrized rectified linear unit (PReLU) [13]. The end result of ResDNet is a noise realization estimate that is subtracted from the distorted image. Before the subtraction takes place, the noise realization is normalized so that its variance matches the input variance. This is accomplished with a trainable ℓ_2-projection layer,

$$\Pi_{\mathcal{C}}(\mathbf{y}) = \theta\mathbf{y}/\max(\|\mathbf{y}\|_2, \theta), \qquad (8)$$

where $\theta = \sigma\sqrt{N-1}$. Overall, this denoising network is relatively small since it contains approximately 380K parameters and it can be easily deployed in each iteration of our INN without requiring excessive memory or computation time.

Algorithm 1: Proposed Iterative Neural Network for bust photography applications

Input: $\mathbf{H}$: Degradation Operator, $\mathbf{y}_{\{1\dots B\}}$: 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 $\mathbf{S}_{1\dots B}$;
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
  $\mathbf{x}^{t+1} = \text{ProxNet}(\mathbf{x}^t - \mathbf{z}/B, \sigma, \boldsymbol{s}_t)$;
end

In order to emphasize that the employed denoising network in our INN serves as a proximal map estimate and not as a single image Gaussian denoiser, hereafter, we will refer to it as ProxNet. Another reason for our naming convention is that our overall approach is not tied to a specific proximal network and in principle ResDNet can be replaced by another network architecture that exhibits similar properties.

5.2. Iterative neural network

The proposed INN combines the PGD algorithm as discussed in Section 4 and the proximal network as an estimator of the solution of Eq. (4). A straightforward way to implement the INN is to use in every iteration a proximal network that is governed by a different set of parameters. However, the training of INN, in this case, becomes quickly intractable, since the number of parameters increases linearly to the number of employed iterations. To deal with this shortcoming, we instead use the same proximal network in every iteration, and thus we keep the number of network parameters small, which in turn decreases the necessary training time and the memory footprint of the network.

In order to speed up the convergence of the optimization scheme, we exploit two commonly used convergence acceleration strategies. The first one is the homotopy continuation strategy [27] where the standard deviation of the noise is deliberately over-estimated in the first iterations and gradually is decreased until the accurate estimation of σ is reached. The homotopy continuation scheme accelerates the convergence of PGD algorithms as shown in [27] and it can be easily integrated into our formulation via a modification of the projection layer by replacing θ with $\hat{\theta} = e^s\theta$. In detail, we initialize the trainable parameter of the projection layer $\boldsymbol{s} \in \mathbb{R}^K$ with values spaced evenly on a log scale from s_{max} to s_{min} and later on the vector $\boldsymbol{s}$ is further finetuned on the training dataset via back-propagation.

The second acceleration strategy that we explore involves the use of an extrapolation step similar to the one introduced in [3]. Specifically, the outputs of two consecutive iterations are combined in a weighted manner in order to obtain the solution of the current iteration. In [3] the extrapolation weights $\mathbf{w} \in \mathbb{R}^K$ are known apriori but in our work, we learn them during the training of INN. We initialize the extrapolation weights as $w_i = \frac{t-1}{t+2}, \forall 1 \leq t \leq K$, which matches the configuration described in [32].

Algorithm 1 describes our overall strategy which combines all the different components that we described, i.e., the PGD, the proximal network, the continuation, and extrapolation strategies. As it can be seen from Algorithm 1, our reconstruction approach has only a weak dependency on the burst size, since this is only involved in the computation of the gradients for each burst observation, which can be done very efficiently. This feature makes our method very efficient since the proximal network is independent to the bust size B, unlike other recent deep learning based methods [11, 1], which process each frame of the burst individually at first and then jointly and therefore the computational time increases linearly to B. Simultaneously, our proposed approach supports by design bursts of arbitrary size with only a minor computational overhead. We note that this is not the case for the network in [31] which is constrained to use bursts of 8 frames. In a different case, the entire network needs to be trained from scratch. Finally, our proposed INN is by definition permutation invariant similarly to [1]. In particular, the ordering of the burst frames does not affect at all the reconstruction result as long as the reference frame remains the same.

6. Network Training

6.1. Synthetic training dataset

Since there are no publicly available burst photography datasets suitable for training our network, we create training pairs of ground-truth and input bursts using the Microsoft Demosaicking Dataset (MSR) [19] for burst image demosaicking and the Waterloo Dataset [28] for burst Gaussian denoising. In both cases, we modify the ground-truth image by affinely transforming it 8 times to create a burst with synthetic misalignment and then the images are center cropped to retain a patch of 128×128 pixel. We assume that the reference frame is the last one and therefore it does not undergo any transformation. The random affine transformation should be close to realistic scenarios, and thus we restrict the transformation to contain a translation in each direction up to 10 pixels and rotation of up to 2 degrees.

For burst image demosaicking, we selected the MSR dataset which is a small but well-known dataset for evalu-

ating image demosaicking algorithms, as explained in [19]. The advantage of the MSR dataset is that all data are in the linear color space where pixel measurements are proportional to the number of counted photons, and no post-processing steps have been performed (e.g., sharpening, tone mapping) that will alter the image statistics. The dataset consists of 200 images for training, 100 for validation and 200 images for testing purposes. For each ground-truth image we generate the respective burst sequence, and then we apply the Bayer pattern on each frame. We also explore the case of noise perturbing the camera measurements, and therefore we add noise sampled from a heteroskedastic Gaussian distribution with signal dependent standard deviation $\hat{\omega} \sim \mathcal{N}(\omega, \alpha\omega + \beta^2)$, following the model presented in [14]. The parameter α is related to the shot noise component, which occurs from the stochastic nature of the photon counting process and it is dependent on the true intensities $\mathbf{y}$, while the parameter β is linked to the signal independent read noise component. Both noise parameters are sampled uniformly from a specific range as discussed in [31], which covers the noise levels of many widely used cameras. The dataset is also augmented with random flipping and color jittering in order to ensure a plethora of lighting conditions.

For burst image denoising, we use the Waterloo dataset which consists of 4,744 images. Using the described procedure, we retrieved the synthetically mis-aligned bursts of 8 frames and 500 of these bursts were kept separately to be used for testing purposes. All frames were distorted with additive Gaussian noise with standard deviation sampled from $[5, 25]$ with a step size equal to 2.5.

For all experiments, we estimate the warping matrix that aligns every observation to the reference frame using the Enhanced Correlation Coefficient (ECC) [8]. Since the images are severely distorted by noise, we estimate the alignment on the Gaussian pyramid of the image and use the warping matrix of coarse scales to initialize the ECC estimation of finer levels in order to achieve robustness to the noise perturbations. Bursts that failed to be aligned using this method were dropped from the training set.

6.2. Implementation Details

For all experiments we choose the interpolation operation, involved in the affine transformation of the observation model Eq. (1), to be bilinear due to its low computation complexity and the adequate result that it provides. Using a pre-trained proximal network our overall network is further trained end-to-end to minimize the ℓ_1 loss.

Due to the iterative nature of our framework, the network parameters are updated using the Back-Propagation Through Time (BPTT) algorithm, and more specifically we adopt the Truncated BPTT framework presented in [33, 21]. While we unfold K instances of the network, we propagate the gradients through smaller chunks of size k instead of K, due to the inherent memory restrictions we face during training. Every k iterations we update the parameters based on the loss function and then proceed with unrolling the next k iterations till the number of total iterations K is reached. This modification of the standard BPTT allows the usage of larger batch sizes and a higher number of iterations which leads to better performance, as shown in [21]. Furthermore, we set for all experiments $K = 10$, $k = 5$ and the optimization is carried via the AMSGRAD optimizer where the training starts from an initial learning rate which we decrease by a factor of 10 every 100 epochs. The specific hyper-parameters used for training of each model are provided in the supplementary material.

	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

Table 1: PSNR performance of different methods in both linear and sRGB spaces. Every method was tested on both single image and burst scenario. In the case of BM3D-CFA, demosaicking of the denoised images was performed using the noisefree model of [21].

7. Experiments

7.1. Image Demosaicking and Denoising

We evaluate our method on the test set of the burst MSR dataset. In Table 1, we compare our INN with a bilinear interpolation baseline, two recent demosaicking neural networks [10, 21], as well as with a denoising approach using BM3D-CFA [7] followed by demosaicking using the noise-free model of [21]. BM3D-CFA was also used to denoise the raw data for the bilinear interpolation baseline in the noisy scenario. In all comparisons, we consider both a single image scenario and a burst variant where we apply the respective method on each frame of the burst and then the frames are aligned in order to be averaged. Our approach yields substantially better quantitative results than competing methods in both noisy and noise-free scenario with per-

formance gains ranging from 0.9 to 1.5 dB. To visually assess the superiority of our approach, we further provide representative results in Fig. 3.

In order to examine how the alignment of observations affects the results, we have also considered the case where our pre-trained network was fed with oracle warping matrices. As it could be expected, the restoration performance increases up to 1.9 dBs, which highlights the importance of robust image alignment and indicates that we can expect an increase in our network's performance by employing a better alignment method than the one we currently use.

7.2. Gaussian Image Denoising

We tested our method on the Gaussian denoising task where most burst photography methods focus on. For comparisons, we used the methods of BM3D, VBM4D and ResDNet for single and burst scenarios. In the case of the burst variant of ResDNet, the images were first denoised using ResDNet and then aligned using the method [8] before being averaged. For reasons of experimental completeness, we would like to compare our method against the two most recent deep learning approaches [11, 31], however, neither one of the models or their respective testing sets are publicly available yet. From the results presented in Table 2 and Fig. 4, it is clear that our method achieves a state-of-the art performance across every noise level. An interesting result is that our INN, which uses ResDNet as a sub-component, consistently outperforms the burst variant of ResDNet. This is attributed to the principled way we designed our INN so that it faithfully follows the forward model.

We also performed an ablation study on the importance of burst size during training. Specifically, we trained 3 models using bursts of size 2, 4 and 8 and tested them on sequences with burst sizes varying from 2 to 16, as presented in Fig. 2. The models that were trained with 4 and 8 frames are able to generalize well when they are provided with more frames during inference since their performance steadily increases. Nevertheless, there is a performance gap between the models which indicates that the burst size for which the network is originally trained for can affect the performance. The model trained to handle bursts of only two frames exhibits the same behaviour up to a certain number of frames but after that, its performance starts to decline. Our findings contradict the conclusion of the authors in [11] that deep learning models need to be trained with many frames in order to generalize to longer sequences during inference. In fact, our network variants trained for 4 and 8 bursts show a consistent performance improvement with the increase of the burst sequence.

8. Limitations

Our method is capable of producing high-quality images from a burst sequence with great success. However, the main limitation of our network is the dependency it has to the ECC estimation of the warping matrix, which in practice can be rather inaccurate especially when there is a strong presence of noise. When the estimated affine transformation matrix is imprecise, our network inevitably will introduce ghosting artifacts to the final result Fig. 4 (more examples can be found in the supplementary material). In this event, one possible solution, is to estimate the quality of the transformation matrix via a consistency metric like the one in [43] and crop out inconsistent areas from a frame.

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**

Table 2: Color image denoising comparisons for five different noise levels. The restoration quality is measured in terms of average PSNR.

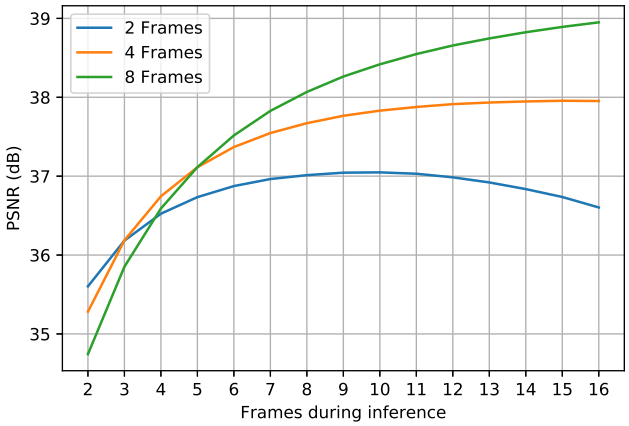

Figure 2: Generalization ability of our INN to different burst sizes. Three models were trained with 2, 4 and 8 frames and tested on burst sequences from 2 to 16 frames.

9. Conclusions

In this work, we have proposed a novel iterative neural network architecture for burst photography applications. Our derived network has been designed to respect the physical model of burst photography while its overall structure is inspired by large-scale optimization techniques. By explicitly taking into account the special characteristics of the problems under study, our network outperforms previous state-of-the-art methods across various tasks, while being invariant to the ordering of the frames and capable to generalize well to arbitrary burst sizes.

Figure 3: Burst demosaicking results on a real and a synthetic burst from the FlexISP dataset [16] (results are best seen magnified on a computer screen). Our model successfully restores the missing colors of the underlying images while suppressing noise. A PSNR comparison of the systems is provided in the supplementary material.

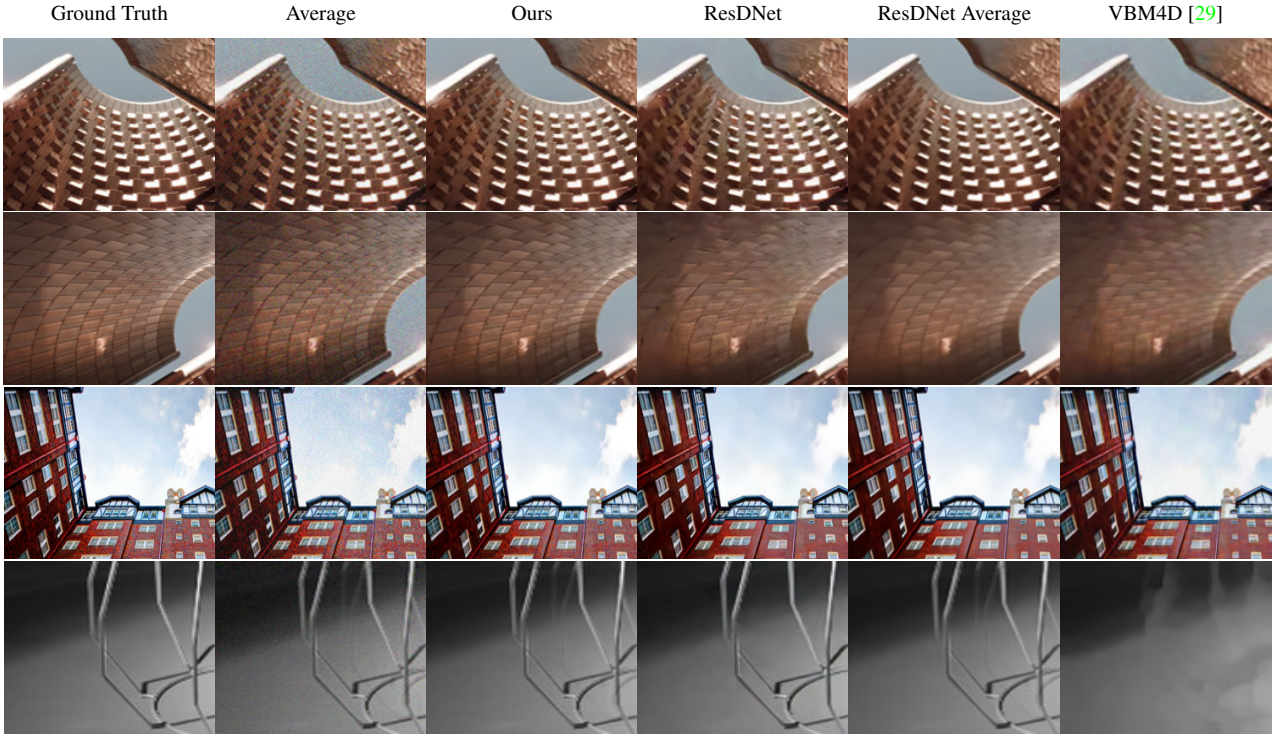

Figure 4: Burst Gaussian denoising with $\sigma = 25$. Our method is able to effectively restore the images and retain fine details, as opposed to the rest of the methods that over-smooth high texture areas. Imprecise misalignment will cause methods to introduce visual artifacts such as those in the last row. Results best seen magnified on a computer screen.

References

[1] Miika Aittala and Fredo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[2] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[3] Amir Beck and Marc Teboulle. A fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[4] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[5] Jian-Feng Cai, Hui Ji, Chaoqiang Liu, and Zuowei Shen. Blind motion deblurring using multiple images. *Journal of Computational Physics*, 228(14):5057 – 5071, 2009.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[7] A. Danielyan, M. Vehvilainen, A. Foi, V. Katkovnik, and K. Egiazarian. Cross-color bm3d filtering of noisy raw data. In *2009 International Workshop on Local and Non-Local Approximation in Image Processing*, pages 125–129, Aug 2009.

[8] G. D. Evangelidis and E. Z. Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, Oct 2008.

[9] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13(10):1327–1344, Oct 2004.

[10] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep Joint Demosaicking and Denoising. *ACM Trans. Graph.*, 35(6):191:1–191:12, Nov. 2016.

[11] Clement Godard, Kevin Matzen, and Matt Uyttendaele. Deep burst denoising. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[12] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T. Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 35(6), 2016.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[14] G. E. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):267–276, March 1994.

[15] Felix Heide, Steven Diamond, Matthias Nießner, Jonathan Ragan-Kelley, Wolfgang Heidrich, and Gordon Wetzstein. Proximal: Efficient image optimization using proximal algorithms. *ACM Transactions on Graphics (TOG)*, 35(4):84, 2016.

[16] Felix Heide, Markus Steinberger, Yun-Ta Tsai, Mushfiqur Rouf, Dawid Pajak, Dikpal Reddy, Orazio Gallo, Jing Liu, Wolfgang Heidrich, Karen Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics (TOG)*, 33(6):231, 2014.

[17] Bernardo Henz, Eduardo S. L. Gastal, and Manuel M. Oliveira. Deep joint design of color filter arrays and demosaicing. *Computer Graphics Forum*, 37(2):389–399, 2018.

[18] K. Hirakawa and T. W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE Transactions on Image Processing*, 14(3):360–369, March 2005.

[19] D. Khashabi, S. Nowozin, J. Jancsary, and A. W. Fitzgibbon. Joint Demosaicing and Denoising via Learned Nonparametric random fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981, Dec 2014.

[20] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[21] Filippos Kokkinos and Stamatios Lefkimmiatis. Iterative residual network for deep joint image demosaicking and denoising. *arXiv preprint arXiv:1807.06403*, 2018.

[22] D Kostadin, F Alessandro, and E KAREN. Video denoising by sparse 3d transform-domain collaborative filtering. In *European signal processing conference*, volume 149. Tampere, Finland, 2007.

[23] Stamatios Lefkimmiatis. Non-local color image denoising with convolutional neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[24] Stamatios Lefkimmiatis. Universal denoising networks : A Novel CNN Architecture for Image Denoising. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] S. Lefkimmiatis, P. Ward, and M. Unser. Hessian Schatten-norm regularization for linear inverse problems. *IEEE Transactions on Image processing*, 22(5):1873–1888, 2013.

[26] Anat Levin and Boaz Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840. IEEE, 2011.

[27] Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Computational Optimization and Applications*, 60(3):633–674, Apr 2015.

[28] Kede Ma, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang. Waterloo Exploration Database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, Feb. 2017.

[29] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising using separable 4d nonlocal spatiotemporal transforms. In *Image Processing: Algorithms and Systems IX*, volume 7870, page 787003. International Society for Optics and Photonics, 2011.

[30] D. Menon and G. Calvagno. Joint demosaicking and denoisingwith space-varying filters. In *2009 16th IEEE Interna-*

tional Conference on Image Processing (ICIP), pages 477–480, Nov 2009.

[31] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018.

[32] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[33] A. J. Robinson and Frank Fallside. The Utility Driven Dynamic Error Propagation Network. Technical Report CUED/F-INFENG/TR.1, Engineering Department, Cambridge University, Cambridge, UK, 1987.

[34] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (RED). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

[35] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 860–867. IEEE, 2005.

[36] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[37] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of International Conference on Computer Vision*, 2017.

[38] A. N. Tikhonov. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.*, 4:1035–1038, 1963.

[39] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-Play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948, Dec 2013.

[40] Lei Zhang Xin Li, Bahadir Gunturk. Image demosaicing: a systematic survey. volume 6822, pages 6822 – 6822 – 15, 2008.

[41] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.

[42] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning Deep CNN Denoiser Prior for Image Restoration. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, July 2017.

[43] Xiaoou Tang Matt Uyttendaele Ziwei Liu, Lu Yuan and Jian Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6), 2014.