Deep Reparametrization of Multi-Frame Super-Resolution and Denoising

Goutam Bhat Martin Danelljan Fisher Yu Luc Van Gool Radu Timofte
Computer Vision Lab, ETH Zurich, Switzerland

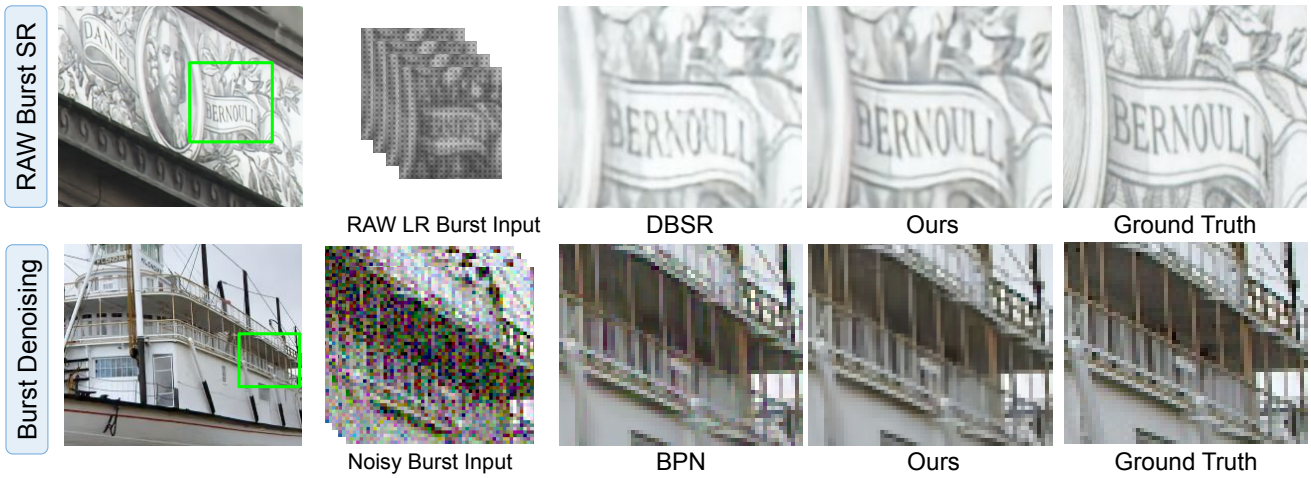

Figure 1. We propose a deep reparametrization of the classical MAP objective (1) for multi-frame image restoration. Our general formulation minimizes a learned reconstruction error in a deep latent space. The proposed approach outperforms previous state-of-the-art methods DBSR [2] and BPN [64] on the RAW burst super-resolution (top) and burst denoising (bottom) tasks, respectively.

Abstract

We propose a deep reparametrization of the maximum a posteriori formulation commonly employed in multi-frame image restoration tasks. Our approach is derived by introducing a learned error metric and a latent representation of the target image, which transforms the MAP objective to a deep feature space. The deep reparametrization allows us to directly model the image formation process in the latent space, and to integrate learned image priors into the prediction. Our approach thereby leverages the advantages of deep learning, while also benefiting from the principled multi-frame fusion provided by the classical MAP formulation. We validate our approach through comprehensive experiments on burst denoising and burst super-resolution datasets. Our approach sets a new state-of-the-art for both tasks, demonstrating the generality and effectiveness of the proposed formulation.

1. Introduction

Multi-frame image restoration (MFIR) is a fundamental computer vision problem with a wide range of important applications, including burst photography [2, 24, 39, 63] and remote sensing [12, 32, 46]. Given multiple degraded and noisy images of a scene, MFIR aims to reconstruct a clean, sharp, and often higher-resolution output image. By effectively leveraging the information contained in different input images, MFIR approaches are able to reconstruct richer details that cannot be recovered from a single image.

As a widely embraced paradigm [15, 17, 36, 47], MFIR is addressed by first modelling the image formation process as, $x_i = H\big(\phi_{m_i}(y)\big) + \eta_i$. In this model, the original image y is affected by the scene motion ϕ_{m_i}, image degradation H, and noise η_i, resulting in the observed image x_i. Assuming the noise η_i follows an i.i.d. Gaussian distribution, the original image y is reconstructed from the set of noisy observations $\{x_i\}_1^N$ by finding the maximum a posteriori (MAP) estimate,

$$\hat{y} = \arg\min_{y} \sum_{i=1}^{N} \left\| x_i - H\left(\phi_{m_i}(y)\right) \right\|_2^2 + \mathcal{R}(y)\,, \qquad (1)$$

where $\mathcal{R}(y)$ is the imposed prior regularization.

While the MAP formulation (1) has enjoyed much popularity, there are several challenges when employing it in real-world settings. The formulation (1) assumes that the degradation operator H is known, which is not often the case. Moreover, it requires manually tuning the regular-

izer $\mathcal{R}(y)$ for good performance [17, 19, 21]. Despite these shortcomings, the MAP formulation (1) provides an elegant modelling of the MFIR problem, and a principled way of fusing information from multiple frames. This inspires us to formulate a deep MFIR method that leverages the compelling advantages of (1), while also benefiting from the end-to-end learning of the degradation operator H and the regularizer $\mathcal{R}$.

We propose a deep reparametrization of the classical MAP objective (1). Our approach is derived as a generalisation of the image space reconstruction problem (1), by transforming the MAP objective to a deep feature space. This is achieved by first introducing an encoder network that replaces the L_2 norm in (1) with a learnable error metric, providing greater flexibility. We then reparametrize the target image y with a decoder network, allowing us to solve the optimization problem in a learned latent space. The decoder integrates strong learned image priors into the prediction, effectively removing the need of a manually designed regularizer $\mathcal{R}$. Our deep reparametrization also allows us to directly learn the effects of complex degradation operator H in the deep latent space of our formulation. To further improve the robustness of our model to *e.g.* varying noise levels and alignment errors, we introduce a network component that estimates the certainty weights of all observations in the objective.

We validate the proposed approach through extensive experiments on two multi-frame image restoration tasks, namely RAW burst super-resolution, and burst denoising. Our approach sets a new state-of-the-art on both tasks by outperforming recent deep learning based approaches (see Fig. 1). We further perform extensive ablative experiments, carefully analysing the impact of each of our contributions.

2. Related Work

Multi-Frame Super-Resolution: MFSR is a well-studied problem, with more than three decades of active research. Tsai and Huang [60] were the first to propose a frequency-domain based solution for MFSR. Peleg *et al*. [47] and Irani and Peleg [31] proposed an iterative approach based on an image formation model. Here, an initial guess of the SR image is obtained and then refined by minimizing a reconstruction error. Several works [1, 15, 22, 51] extended the objective in [31] with a regularization term to obtain a maximum a posteriori (MAP) estimate of the HR image. Robustness to outliers or varying noise levels were further addressed in [17, 70].

The aforementioned approaches assume that the image formation model, as well as the motion between input frames can be reliably estimated. Several works address this limitation by jointly estimating these unknown parameters [16, 26, 34, 48, 68], or marginalizing over them [48, 49, 58]. Alternatively, a number of approaches directly predict the HR image without simulating the image formation process. Chiang and Boult [9] upsample and warp the input images to a common reference, before fusing them. Farsiu *et al*. [18] extend this approach with a robust regularization term. Takeda *et al*. [55, 56] proposed a kernel regression based approach for super-resolution. Wronski *et al*. [63] used the kernel regression technique to perform joint demosaicking and super-resolution. A few deep learning based solutions have also been proposed recently for MFSR, mainly focused on remote sensing applications [12, 32, 46]. Bhat *et al*. [2] propose a learned attention-based fusion approach for hand held burst super-resolution. Haris *et al*. [23] propose a recurrent back-projection network for video super-resolution.

Multi-Frame Denoising: In addition to the MFSR approaches discussed previously, a number of specialized multi-frame denoising approaches have also been proposed in the literature. Tico [57] performs block matching both within an image, as well as across the input images to perform denoising. [10, 41, 42] extend the popular image denoising algorithm BM3D [11] to video. Buades *et al*. [7] estimate the noise level from the aligned images, and use a combination of pixel-wise mean and BM3D to denoise. Hasinoff *et al*. [24] used a hybrid 2D/3D Wiener filter to denoise and merge burst images for HDR and low-light photography applications. Godard *et al*. [20] extend a single frame denoising network for multiple frames using a recurrent neural network. Mildenhall *et al*. [45] employed a kernel prediction network (KPN) to obtain per-pixel kernels which are used to merge input images. The KPN approach was then extended by [43] to predict multiple kernels, while [64] introduced basis prediction networks to enable the use of larger kernels.

Deep Optimization-based image restoration: A number of deep learning based approaches [35, 36, 65, 66] have posed image restoration tasks as an explicit optimization problem. The P^3 [61] and RED [50] approaches provide a general framework for utilizing standard denoising methods as regularizers in optimization-based image restoration methods. Zhang *et al*. [66] used the half quadratic splitting method to plug a deep neural-network based denoiser prior into model-based optimization methods. Kokkinos *et al*. [36] used a proximal gradient descent based framework to learn a regularizer network for burst photography applications. These prior works mainly focus on only learning the regularizer, while assuming that the data term (image formation process) is known and simple. Furthermore, the reconstruction error computation, as well as the error minimization are restricted to be in the image space. In contrast, our deep reparametrization approach allows jointly learning the imaging process as well as the priors, without restricting the image formation model to be simple or linear.

3. Method

3.1. Problem formulation

In this work, we tackle the multi-frame (MF) image restoration problem. Given multiple images $\{x_i\}_{i=1}^N$, $x_i \in \mathbb{R}^{h\times w\times c_{\text{in}}}$ of a scene, the goal is to merge information from these input images to generate a higher quality output $\hat{y} \in \mathbb{R}^{sh\times sw\times c_{\text{out}}}$. Here, c_{in} and c_{out} are the number of image channels, while s is the super-resolution factor. We consider a general scenario where the input images are either captured using a stationary or a hand held camera. The input and output images can either be in RAW or RGB format, depending on the end application.

One of the most successful paradigms to MF restoration and super-resolution in the literature [1, 17, 22, 47] is to first model the image formation process,

$$x_i = H\big(\phi_{m_i}(y)\big) + \eta_i \tag{2}$$

Here, y is the underlying image and ϕ_{m_i} is the warping operation which accounts for scene motion m_i. The image degradation operator H models *e.g.* camera blur, and the sampling process in camera. The observation noise $\eta_i \sim p_\eta$ is assumed to follow a given distribution p_η. The degradation operator H and the scene motion m_i take different forms depending on the addressed task. For example, in the super-resolution task, H acts as the downsampling kernel. Similarly, the scene motion m_i can denote the parameters of an affine transformation, or represent a per-pixel optical flow in case of dynamic scenes. Note that the degradation operator H as well as the scene motion m_i are unknown in general and need to be estimated.

Given the imaging model (2), the original image y is generally estimated by minimizing the error between each observed image x_i, and its simulated counterpart $\bar{x}_i = H\left(\phi_{m_i}(y)\right)$, using the maximum a posteriori (MAP) estimation technique. If the observation noise follows an i.i.d. Gaussian distribution, the MAP estimate $\hat{y}$ is obtained as,

$$\hat{y} = \arg\min_y \sum_{i=1}^{N} \left\| x_i - H\left(\phi_{m_i}(y)\right) \right\|_2^2 + \mathcal{R}(y) \tag{3}$$

Here, $\mathcal{R}(y)$ is the regularization term that integrates prior knowledge about the original image y.

The formulation (3) provides a principled way of integrating information from multiple frames, leading to its popularity. However, it requires manually tuning the degradation operator H and the regularizer $\mathcal{R}$, while also lacking the flexibility to generalize to more complex noise distributions. In this work, we propose a deep reparametrization of (3) to address the aforementioned issues.

3.2. Deep Reparametrization

We introduce a deep reparametrization that transforms the optimization problem (3) into a learned deep feature space (see Figure 2). In this section, we will first derive our approach based on the reconstruction loss (3), and then discuss its advantages over the original image-space formulation. Our generalized deep image reconstruction objective is derived from (3) in three steps, detailed next.

Step 1: We note that the first term in problem (3) minimizes a L_2 distance $\|x_i - \bar{x}_i\|_2$ between the observed image x_i and the simulated image $\bar{x}_i = H\left(\phi_{m_i}(y)\right)$. Instead of limiting the objective to the squared error $\|x_i - \bar{x}_i\|_2^2$ in image space, we learn a more general distance measure $d(x_i, \bar{x}_i)$. We parametrize the metric d by an encoder network E, to obtain image embeddings $E(x_i) \in \mathbb{R}^{\tilde{h}\times\tilde{w}\times c_{\text{e}}}$. The error $d(x_i, \bar{x}_i)$ is then computed as the L_2 distance between the embeddings of the input image x_i and the simulated image $\bar{x}_i$, as $d(x_i, \bar{x}_i) = \|E(x_i) - E(\bar{x}_i)\|_2$. Thanks to the depth and non-linearity of the encoder E, the distance measure d can represent highly flexible error metrics, more suitable for complex noise and error distributions.

Step 2: While the encoder E maps the error computation to a deep feature space, the resulting objective is still minimized in the output image space y. As a second step, we therefore reparametrize the objective (3) in terms of a latent deep representation $z \in \mathbb{R}^{\tilde{s}\tilde{h}\times\tilde{s}\tilde{w}\times c_z}$ of the image y. To this end, we introduce a decoder network D that maps the latent representation z to the estimated image $y = D(z)$. Since z is a direct parametrization of the target image y, we can optimize the objective w.r.t. z and predict the final image as $\hat{y} = D(\hat{z})$ once the optimal latent representation $\hat{z}$ is found. The resulting objective is thus expressed as,

$$L(z) = \sum_{i=1}^{N} \left\| E(x_i) - E \circ H \circ \phi_{m_i} \circ D(z) \right\|_2^2 + \mathcal{R}\left(D\left(z\right)\right)$$
$$\hat{y} = D(\hat{z})\,, \quad \hat{z} = \arg\min_z L(z)\,. \tag{4}$$

Here, '$\circ$' denotes composition $f \circ g(\cdot) = f(g(\cdot))$ of two functions f, g.

Next, we assume the decoder D to be equivariant w.r.t. the warping operation ϕ_{m_i}. That is, the decoder and warping operation commute as $\phi_{m_i} \circ D = D \circ \phi_{m_i}$. In fact, if ϕ_{m_i} is solely composed of a translation, this condition is readily ensured by the translational equivariance of the CNN decoder D. For more complex motions, the equivariance condition still holds to a good approximation if the motion m_i locally resembles a translation. This is generally the case for the considered burst photography settings, where the motion between frames are small to moderate. Furthermore, as for optical flow networks that also employ feature warping [29, 54], our decoder D can learn to accommodate the desired warping equivariance through end-to-end training. By using the equivariance condition $\phi_{m_i} \circ D = D \circ \phi_{m_i}$

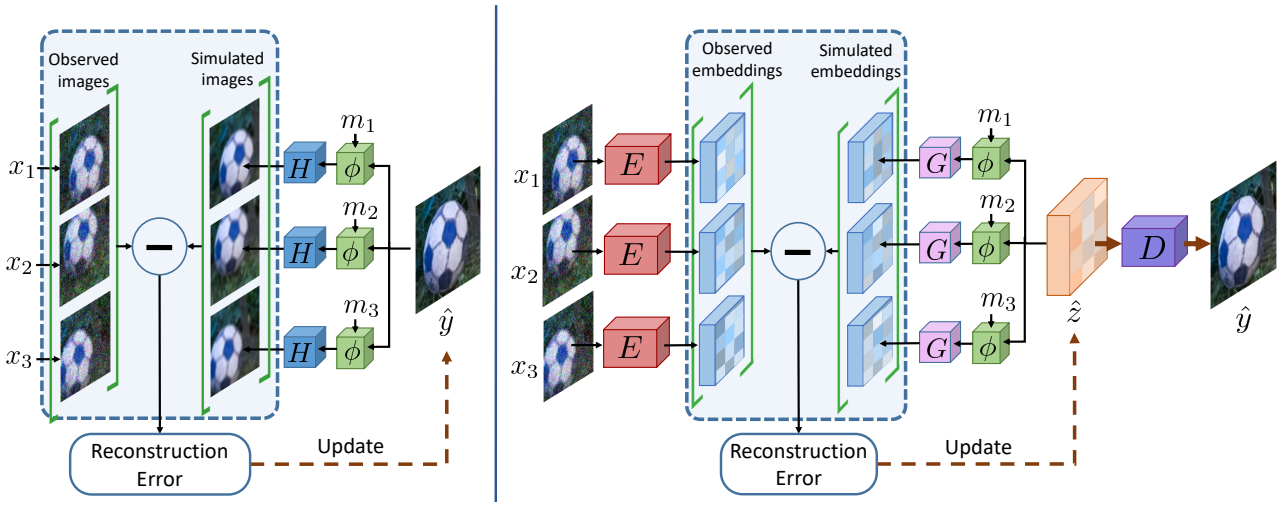

Figure 2. **Left**: Classical multi-frame image restoration approaches minimize a reconstruction error (3) between the observed images x_i and the simulated images $H\left(\phi_{m_i}\left(\hat{y}\right)\right)$ to obtain the output image $\hat{y}$. **Right**: In contrast, we employ an encoder E to compute the reconstruction error (6) in a learned feature space. The reconstruction error is minimized w.r.t. a latent representation z, which is then passed through the decoder D to obtain the prediction $\hat{y}$.

in (4), we obtain the objective,

$$L(z)=\sum_{i=1}^{N}\left\|E(x_i)-\underbrace{E\circ H\circ D}_{G}\circ\phi_{m_i}(z)\right\|_2^2+\mathcal{R}\left(D\left(z\right)\right), \tag{5}$$

which allows us to directly apply the warping ϕ_{m_i} on the latent representation z.

Step 3: As the final step, we focus on the degradation operator H. In general, H is unknown and thus needs to be estimated or learned. Although it could be directly parametrized as a separate neural network, we propose a different strategy. By directly comparing (3) and (5), we interestingly find the role of H in (3) replaced by the composition $G = E\circ H\circ D$ in (5). Instead of learning the image space degradation map H, we can thus directly parametrize its resulting deep feature space operator G. Here, G can be seen as the feature space degradation operator, which is used to directly obtain the simulated image embedding $G\left(\phi_{m_i}(z)\right)$. We thereby obtain the following objective,

$$L(z)=\sum_{i=1}^{N}\left\|E(x_i)-G\big(\phi_{m_i}(z)\big)\right\|_2^2+\mathcal{Q}(z)\,. \tag{6}$$

In (6), we have also introduced the latent space regularizer $\mathcal{Q}=\mathcal{R}\circ D$, which can similarly be parametrized directly in order to avoid invoking the decoder D during the optimization process. Next, we will discuss the advantages of our deep reformulation (6) of (3), brought by each of the neural network modules E, D, and G.

Encoder E: The encoder maps the input images x_i to an embedding space $E(x_i)$, where the reconstruction error is defined. It can thus learn to transform complex noise distributions p_η and other error sources, stemming from *e.g.* inaccurate motion estimation m_i, to a feature space where it is better approximated as independent Gaussian noise. Our approach thus avoids strict assumptions imposed by the L_2 loss in image space through the flexibility of the encoder E.

Decoder D: The minimization problem (3) is often solved using iterative numerical methods, such as the conjugate gradient method. The convergence rate of such methods strongly depend on the conditioning of the objective. Since we optimize (5) w.r.t. a latent representation z instead of the output image y, our decoder D serves as a preconditioner, leading to faster convergence. Furthermore, while effective image space regularizers $\mathcal{R}(y)$ are often complex [17, 21], our latent parametrization z allows for trivial regularizers $\mathcal{Q}(z)$. Similar to CNN-based single-image super-resolution approaches [13, 38, 40, 69], the decoder D also learns strong image priors which are applied during the prediction step $\hat{y} = D(\hat{z})$. Thanks to the regularizing effect of our decoder, we found it sufficient to simply set $\mathcal{Q}(z) = \lambda\|z\|_2^2$ where λ is a learnable scalar.

Feature degradation G: The image degradation operator H can be complex and non-linear in general, making it hard to solve the minimization problem (3). In our deep reformulation (6) of (3), the image degradation H is replaced by its feature space counterpart $G = E\circ H\circ D$. Here, the encoder E and decoder D are deep neural networks, capable of learning highly non-linear mappings. These can therefore learn a latent space where the degradation operation is approximately linear. That is, for a given image degradation H, we can learn appropriate G, E and D such that $G\approx E\circ H\circ D$ even in the case when G is constrained to be linear. Consequently, we constrain G to be a linear convolution filter, which is accommodated by the end-to-end learning of suitable E and D where such a linear relation holds.

As a result, our optimization problem (6) is convex and can be easily optimized using efficient quadratic solvers.

We model the encoder E, decoder D, and degradation G as convolutional neural networks. As detailed in Sec. 3.5, these networks are learned directly from data. But first, we propose a further generalization to our objective (6).

3.3. Certainty Predictor

In our formulation (6), the reconstruction error for each frame, location, and feature channel are weighted equally. This is the correct model if the errors, often seen as observation noise, are identically distributed. In practice however, images are affected by heteroscedastic noise [27], which varies spatially depending on the image intensity value. Furthermore, the reconstruction errors in (3) and (6) are affected by the quality of the motion estimation m_i. In practical applications, the scene motion m_i is unknown and needs to be estimated using *e.g.* optical flow. As a result, the estimated m_i may contain significant errors for certain regions, leading to sub-optimal results. In order to model these effects, we further introduce a certainty predictor module W.

Our certainty predictor aims to determine element-wise certainty values $v_i \in \mathbb{R}^{\tilde{h}\times\tilde{w}\times c_{\text{e}}}$ for each element in the residual $E(x_i) - G\left(\phi_{m_i}\left(z\right)\right)$. Intuitively, image regions with higher noise or unreliable motion estimate m_i should be given lower certainty weights, effectively reducing their impact in the MAP objective (6). The certainty values v_i are computed using the image embeddings $\{E(x_j)\}_{j=1}^N$, motion estimate m_i, and the noise level n_i (if available) as input. Our final optimization problem, including the certainty weights v_i is then expressed as,

$$L(z) = \sum_{i=1}^{N} \left\| v_i \cdot \left(E(x_i) - G\left(\phi_{m_i}\left(z\right)\right)\right) \right\|_2^2 + \lambda \|z\|_2^2 \quad \text{where} \quad v_i = W\left(\{E(x_j)\}_{j=1}^N, m_i, n_i\right). \tag{7}$$

In relation to the MAP estimation (3), the certainty weights correspond to an estimate of the inverse standard deviation $v_i = \frac{1}{\sigma_i}$ of the encoded observations $E(x_i)$.

3.4. Optimization

To ensure practical inference and training, it is crucial that our objective (7) can be minimized efficiently. Furthermore, in order to learn our network components end-to-end, the optimization solver itself needs to be differentiable. Due to the linearity of warp operator ϕ_{m_i} and the choice of linear feature degradation G, our objective $L(z)$ is a linear least-squares problem, which can be addressed with standardized techniques. In particular, we employ the steepest-descent algorithm, which can be seen as a simplification of the Conjugate Gradient [52]. Both algorithms have been previously employed in classical MFIR approaches [1, 22], and more recently in deep optimization-based few-shot learning approaches [3, 4, 59].

The steepest-descent algorithm performs an optimal line search $\alpha^j = \arg\min_\alpha L(z^j - \alpha g^j)$ in the gradient $g^j = \nabla L(z^j)$ direction to update the iterate $z^{j+1} = z^j - \alpha^j g^j$. Since the problem is quadratic, simple closed-form expressions can be derived for both the gradient g^j and step length α^j. For our model (7), the complete algorithm is given by,

$$\begin{aligned} g^j &= -2\sum_{i=1}^{N} \phi_{m_i}^{\text{T}} G *^{\text{T}} \left(v_i^2 \cdot \left(E(x_i) - G * \phi_{m_i}(z^j)\right)\right) + 2\lambda z^j \\ \alpha^j &= \frac{\|g^j\|_2^2}{\sum_{i=1}^{N} 2\|v_i \cdot \left(G * \phi_{m_i}\left(g^i\right)\right)\|_2^2 + 2\lambda\|g^j\|_2^2} \\ z^{j+1} &= z^j - \alpha^j g^j \,. \end{aligned} \tag{8}$$

Here, $*$, $*^{\text{T}}$, and $\cdot$ denote the convolution, transposed convolution, and element-wise product, respectively. Further, $\phi_{m_i}^{\text{T}}$ is the transposed warp operator. A detailed derivation is provided in the supplementary material. Note that both the gradient g^j and step length α^j can be implemented using standard differentiable neural network operations.

To further improve convergence speed, we learn an initializer P which predicts the initial latent encoding $z^0 = P(E(x_1))$ using the embedding of the first image x_1. Our approach then proceeds by iteratively applying K_{SD} steepest-descent iterations (8). Due to the fast convergence provided by the steepest-descent steps, we found it sufficient to only use $K_{\text{SD}} = 3$ iterations. By unrolling the iterations, our optimization module can be represented as a feed-forward network $A_{G,W,P}$ that predicts the optimal encoding $\hat{z}$. Our complete inference procedure is then expressed as,

$$\hat{y} = D\left(A_{G,W,P}\left(\left\{\left(E(x_i), m_i\right)\right\}_{i=1}^N\right)\right) \tag{9}$$

In the next section, we will describe how all the components in our architecture can be directly learned end-to-end.

3.5. Training

Our entire MFIR network is trained end-to-end from data in a straightforward manner, without enforcing any additional constraints on the individual components. We use a training dataset $\mathcal{D} = \{(\{x_i^k\}_{i=1}^N, y^k)\}$ consisting of input-target pairs. For each input $\{x_i^k\}_{i=1}^N$, we obtain the prediction $\hat{y}^k$ using (9). The network parameters for each of our components E, G, W, P, and D are then learned by minimizing a prediction error $\ell(y^k, \hat{y}^k)$ over the training dataset $\mathcal{D}$ using *e.g.* stochastic gradient descent. In this work, we use the popular L_1 loss $\ell(y, \hat{y}) = \|y - \hat{y}\|_1$.

4. Applications

We describe the application of our approach to RAW burst super-resolution, and burst denoising tasks. A detailed description is provided in the supplementary material.

4.1. RAW Burst Super-resolution

Here, the method is given a set of RAW bayer images captured successively from a hand held camera. The task is to exploit these multiple shifted observations to generate a denoised, demosaicked, higher-resolution output. In this setting, the image degradation H can be seen as a composition of camera blur, decimation, sampling, and mosaicking operations. Next, we briefly describe our architecture.

Encoder E: The encoder packs each 2×2 block in the input RAW image along the channel dimension to obtain a 4 channel input. This is then passed through an initial conv. layer followed by a series of residual blocks [25] with ReLU activations and without BatchNorm [30]. A final conv. layer predicts a 256-dimensional encoding of the input image.

Operator G: We use a conv. layer with stride $\tilde{s}$ as our feature-space degradation G. The stride $\tilde{s}$ corresponds to the downsampling factor of G. Note that this downsampling need not be the same as the downsampling factor s of the image degradation H. Our latent representation z can encode higher-resolution information in the channel dimension, enabling use of a smaller $\tilde{s}$ for efficiency. We empirically observed that it is sufficient to set $\tilde{s} = 2$ and perform the remaining upsampling by factor $s/\tilde{s}$ in our decoder D.

Decoder D: Our decoder consists of a series of residual blocks (same type as in E), followed by upsampling by a factor of $s/\tilde{s}$ using sub-pixel convolution [53]. The upsampled feature map is passed through additional residual blocks, followed by a final conv. layer to obtain $\hat{y}$.

Motion Estimation: We compute the motion m_i between each input image x_i and a reference image x_1 as pixel-wise optical flow in order to be robust to small object motions in the scene. Specifically, we use a PWCNet [54] trained by the authors on the synthetic FlyingChairs [14], FlyingThings3D [44], and MPI Sintel [8] datasets.

Certainty predictor W: We uses three sources of information in order to predict the certainty v_i: i) The encoding $E(x_i)$ which provides information about local image structure *e.g.* presence of an edge, texture *etc.* ii) The residual $E(x_i) - \phi_{m_i}(E(x_1))$ between the encoding of i-th image x_i and the reference image encoding $E(x_1)$ warped to i-th image, which can indicate possible alignment failures, and iii) The sub-pixel sampling location m_i mod 1 of the pixels in the i-th image. These three entities are passed through a residual network to obtain the certainty v_i for image x_i.

4.2. Burst Denoising

Given a burst of noisy images, the aim of burst denoising is to generate a clean output image. In general, burst denoising requires filtering over both the temporal and spatial dimensions. While the classical MAP formulation (3) accommodates the latter by specially designed regularizers, our approach can learn spatial filtering through two mechanisms. First, the encoder E and decoder D networks allow effective spatial aggregation. Second, our certainty predictor can predict both frame-wise and spatial (through channel-dimension encodings) aggregation weights. Following [43, 45, 64], we consider a burst denoising scenario where an estimate of per-pixel noise variance n_i is available. In practice, such an estimate is available from the exposure parameters reported by the camera. Next, we briefly detail our network architecture employed for this task.

Encoder E: We concatenate the image x_i and the noise estimate n_i and pass it through a residual network to obtain the noise conditioned image encodings $E(x_i, n_i)$

Operator G: We use a conv. layer as our operator G.

Decoder D: Our decoder consists of a series of residual blocks, followed by a final conv. layer which outputs $\hat{y}$.

Motion Estimation: We use a similar strategy as employed in Sec. 4.1 to estimate the motion between images.

Certainty predictor W: We use a similar certainty predictor as employed in Sec. 4.1, with a minor modification. We input the noise estimate n_i directly to W as to condition our minimization problem (7) on the input noise level.

5. Experiments

We perform comprehensive evaluation of our approach on RAW burst super-resolution and burst denoising tasks. Detailed results are provided in the supplementary material.

5.1. RAW Burst Super-Resolution

Here, we evaluate our approach on the RAW burst super-resolution task. Our experiments are performed on the SyntheticBurst dataset, and the BurstSR dataset, both introduced in [2]. The SyntheticBurst dataset consists of synthetically generated RAW bursts, each containing 14 images. The bursts are generated by applying random translations and rotations to a sRGB image, and converting the shifted images to RAW format using an inverse camera pipeline [5]. The BurstSR dataset, on the other hand, contains real-world bursts captured using a hand held smartphone camera, along with a high-resolution ground truth captured using a DSLR camera. Since the input bursts and HR ground truth are captured using different cameras, there are spatial and color mis-alignments between the two, posing additional challenges for both training and evaluation. We perform super-resolution by a factor $s = 4$ in all our experiments.

Training details: For evaluation on the SyntheticBurst dataset, we train our model on synthetic bursts generated using sRGB images from the Zurich RAW to RGB [28] training set. We use a fixed burst size $N = 14$ during our training. Our model is trained for 500k iterations, with a batch size of 16, using the ADAM [33] optimizer. The model trained on the synthetic data is then additionally fine-tuned for 40k iterations on the BurstSR training set for

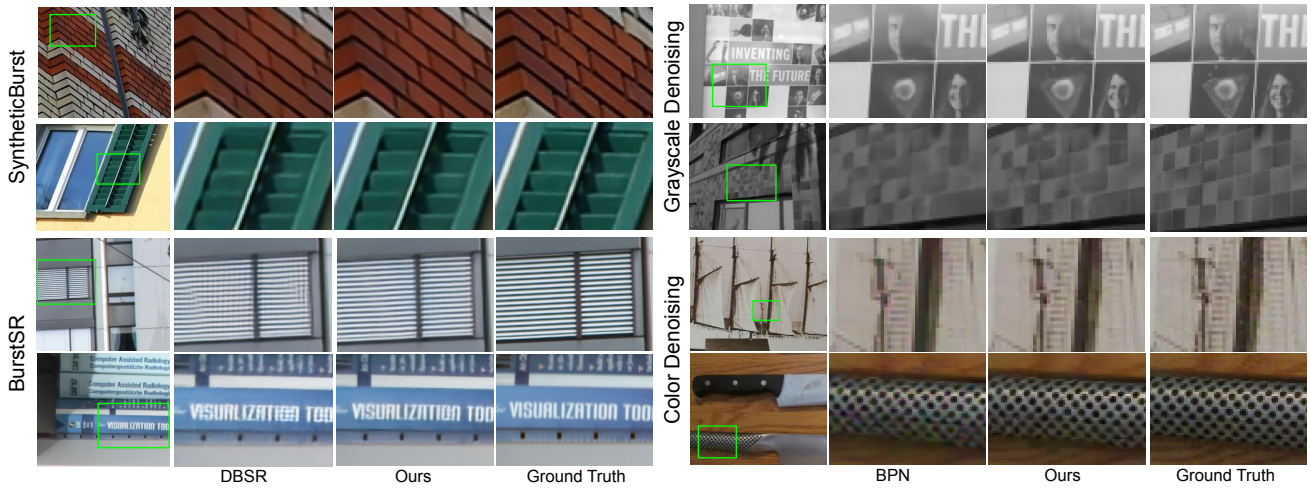

Figure 3. Qualitative comparison of our approach with the previous state-of-the-art methods DBSR [2] and BPN [64] on RAW burst super-resolution (first four columns) and burst denoising (last four columns) tasks.

	SyntheticBurst			BurstSR			
	PSNR↑	LPIPS ↓	SSIM↑	PSNR↑	LPIPS ↓	SSIM↑	Time (s)
SingleImage	36.86	0.113	0.919	46.60	0.039	0.979	**0.02**
HighResNet [12]	37.45	0.106	0.924	46.64	0.038	0.980	**0.11**
DBSR [2]	**40.76**	**0.053**	**0.959**	**48.05**	**0.025**	**0.984**	0.24
Ours	**41.56**	**0.045**	**0.964**	**48.33**	**0.023**	**0.985**	0.40

Table 1. Comparison on the SyntheticBurst and real-world BurstSR validation dataset from [2].

evaluation on the BurstSR val set. In order to handle the mis-alignments between the input and the ground truth in BurstSR dataset, we perform a spatial and color alignment of the network prediction to the ground truth, using the strategy employed in [2], before computing the prediction error.

Results: We compare our approach with the recently introduced DBSR [2] that employs a deep network with an attention-based fusion of input images. Our approach employs the same optical flow estimation network as DBSR. We also compare with HighResNet [12], and a CNN-based single-image baseline consisting of only our encoder and decoder modules. All models are trained using the same training settings as our approach, and evaluated using all available burst images ($N = 14$). The results on the SyntheticBurst dataset containing 300 bursts, in terms of PSNR, SSIM [62], and LPIPS [67] are shown in Tab. 1. All metrics are computed in linear image space. Our approach, minimizing a feature-space reconstruction error, obtains the best results, outperforming DBSR by $+0.80$ dB in PSNR. We also report results on the real-world BurstSR val set containing 882 bursts, using the evaluation strategy described in [2] to handle the spatial and color mis-alignments. Our approach obtains promising results, outperforming DBSR by $+0.28$ dB in PSNR. These results demonstrate that our deep reparametrization of the classical MAP formulation generalizes to real-world degradation and noise. The computation time required to process a burst containing 14 RAW images to generate a 1896 × 1080 RGB output is also reported in Tab. 1. A qualitative comparison is provided in Fig. 3.

	Gain $\propto 1$	Gain $\propto 2$	Gain $\propto 4$	Gain $\propto 8$	Average
HDR+ [24]	31.96	28.25	24.25	20.05	26.13
BM3D [11]	33.89	31.17	28.53	25.92	29.88
NLM [6]	33.23	30.46	27.43	23.86	28.75
VBM4D [42]	34.60	31.89	29.20	26.52	30.55
SingleImage	35.16	32.27	29.34	25.81	30.65
KPN [45]	36.47	33.93	31.19	27.97	32.39
MKPN [43]	36.88	34.22	31.45	28.52	32.77
BPN [64]	38.18	35.42	32.54	**29.45**	33.90
Ours	**39.37**	**36.51**	**33.38**	**29.69**	**34.74**
Ours†	**39.10**	**36.14**	**32.89**	28.98	**34.28**

Table 2. Comparison of our method with prior approaches on the grayscale burst denoising set [45] in terms of PSNR. Results for the first four methods are from [45], while the results for MKPN are from [64]. Our approach obtains the best results, outperforming the previous state-of-the-art method BPN on all noise levels.

5.2. Burst Denoising

We evaluate our approach on the grayscale and color burst denoising datasets introduced in [45] and [64], respectively. Both datasets are generated synthetically by applying random translations to a base image. The shifted images are then corrupted by adding heteroscedastic Gaussian noise [27] with variance $\sigma_r^2 + \sigma_s x$. Here x is the clean pixel value, while σ_r and σ_s denote the read and shot noise parameters, respectively. During training, the noise parameters $(\log(\sigma_r), \log(\sigma_s))$ are sampled uniformly in the log-domain from the range $\log(\sigma_r) \in [-3, -1.5]$ and $\log(\sigma_s) \in [-4, -2]$. The networks are then evaluated on 4 different noise gains $(1, 2, 4, 8)$, corresponding to noise parameters $(-2.2, -2.6)$, $(-1.8, -2.2)$, $(-1.4, -1.8)$, and $(-1.1, -1.5)$, respectively. Note that the noise parameters for the highest noise gain (Gain $\propto 8$) are unseen during training. Thus, performance on this noise level can indicate the generalization of the network to unseen noise. The noise parameters $(\log(\sigma_r), \log(\sigma_s))$ are assumed to be known both during training and testing, and can be utilized to estimate per-pixel noise variance.

Training details: Following [45], we use the images from

	Gain ∝ 1	Gain ∝ 2	Gain ∝ 4	Gain ∝ 8	Average	Runtime (s)
SingleImage	37.94	34.98	31.74	28.03	33.17	**0.005**
KPN [45]	38.86	35.97	32.79	30.01	34.41	-
BPN [64]	40.16	37.08	33.81	31.19	35.56	0.328
Ours	**42.21**	**39.13**	**35.75**	**32.52**	**37.40**	0.198
Ours†	**41.90**	**38.85**	**35.48**	**32.29**	**37.13**	**0.046**

Table 3. Comparison with previous methods on the color burst denoising set [64] in terms of PSNR. The results for KPN are from [64]. Our approach outperforms BPN on all four noise levels.

the Open Images [37] training set to generate synthetic bursts. We train on bursts containing $N = 8$ images with resolution 128×128. Our networks are trained using the ADAM [33] optimizer for 150k and 300k iterations for the grayscale and color denoising tasks, respectively. The entire training takes less than 40h on a single Nvidia V100 GPU.

Results: We compare our approach with the recent kernel prediction based approaches KPN [45], MKPN [43], and BPN [64]. Since our motion estimation network (PWCNet) is trained on external synthetic data, we include a variant of our approach, denoted as Ours†, using a custom optical flow network. Our flow network is jointly trained with the rest of the architecture using a photometric loss, without any extra supervision or data. We also include results for the popular denoising algorithms [6, 11, 42] based on non-local filtering, the multi-frame HDR+ method [24], as well as a single image baseline consisting of only our encoder and decoder. The results over the 73 bursts from the grayscale burst denoising dataset [45], are shown in Tab. 2. Our approach sets a new state-of-the-art, outperforming the previous best method BPN [64] on all four noise levels. Ours† employing a custom flow network also obtains promising results, outperforming BPN on three out of four noise levels.

We also evaluate our approach on the recently introduced color burst denoising dataset [64] containing 100 bursts. The results, along with the computation time for processing a 1024×768 resolution burst, are shown in Tab. 3. Further qualitative comparison is provided in Fig. 3. As in the grayscale set, our approach obtains the best results, significantly outperforming the previous best method BPN. Ours† employing a custom flow network also outperforms BPN by over 1.5 dB in average PSNR, while operating at a significantly higher speed. Furthermore, note that unlike BPN and KPN that are restricted to operate on fixed-size bursts, our approach can operate with bursts of any size, providing additional flexibility for practical applications.

5.3. Ablation Study

Here, we analyse the impact of key components in our formulation. The experiments are performed on the SyntheticBurst super-resolution dataset [2] and the grayscale burst denoising dataset [45]. We train different variants of our approach, with and without the encoder E, decoder D, and the certainty predictor W. This is achieved by replacing the encoder/decoder by an identity function, and setting certainty weights v_i to all ones, when applicable. In order

	E	D	W	SyntheticBurst PSNR	SyntheticBurst ΔPSNR	Denoising PSNR	Denoising ΔPSNR
(a)				31.91	-7.91	28.06	-6.68
(b)			✓	33.85	-5.97	33.00	-1.74
(c)	✓			36.71	-3.11	33.46	-1.28
(d)		✓		38.12	-1.70	32.99	-1.75
(e)	✓	✓		38.36	-1.46	34.54	-0.20
(f)	✓		✓	38.44	-1.38	33.69	-1.05
(g)		✓	✓	38.63	-1.19	34.56	-0.18
(h)	✓	✓	✓	39.82		34.74	

Table 4. Impact of our encoder E, decoder D, and certainty predictor W modules on SyntheticBurst [2] and grayscale denoising [45] datasets. ΔPSNR denotes difference with our final model (h).

to ensure fairness, we employ a deeper decoder when not utilizing an encoder, and vice versa. For training on the SyntheticBurst dataset, we employ a shorter training schedule with 100k iterations. The mean PSNR on the SyntheticBurst set, as well as the mean PSNR over all four noise levels in the grayscale denoising set are provided in Tab. 4.

Minimizing the reconstruction error directly in the input image space (MAP estimate (3)) leads to poor results on both super-resolution and denoising tasks (a). Note that unlike in the classical MAP based approaches, the degradation operator H is still learned in this case. The performance of the image space formulation is improved by employing our certainty predictor (b). The improvement is more prominent in the burst denoising task, where the certainty values allow handling varying noise levels. Our variants employing only the encoder (c), or the decoder (d) module obtain better performance, thanks to the increased modelling capability provided by the use of deep networks. The certainty predictor W provides additional improvements, even when employed together with the encoder (f) or the decoder (g). Removing either of the three components E, D, or W (g)-(e) from our final version leads to a decrease in performance, demonstrating that each of these components are crucial. The performance decrease is much larger in the super-resolution task due to the more complex image degradation process.

6. Conclusion

We propose a deep reparametrization of the classical MAP formulation for multi-frame image restoration. Our approach minimizes the MAP objective in a learned deep feature-space, w.r.t. a latent representation of the output image. Crucially, our deep reparametrization allows learning complex image formation processes directly in latent space, while also integrating learned image priors into the prediction. We further introduce a certainty predictor module to provide robustness to *e.g.* alignment errors. Our approach obtains state-of-the-art results on RAW burst super-resolution as well as burst denoising tasks.

Acknowledgments: This work was supported by a Huawei Technologies Oy (Finland) project, the ETH Zürich Fund (OK), an Amazon AWS grant, and Nvidia.

References

[1] B. Bascle, A. Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In *ECCV*, 1996. 2, 3, 5

[2] Goutam Bhat, Martin Danelljan, L. Gool, and R. Timofte. Deep burst super-resolution. In *CVPR*, 2021. 1, 2, 6, 7, 8

[3] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6182–6191, 2019. 5

[4] Goutam Bhat, Felix Järemo Lawin, Martin Danelljan, Andreas Robinson, Michael Felsberg, Luc Van Gool, and Radu Timofte. Learning what to learn for video object segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020. 5

[5] T. Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and J. Barron. Unprocessing images for learned raw denoising. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11028–11037, 2019. 6

[6] A. Buades, B. Coll, and J. Morel. A non-local algorithm for image denoising. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2:60–65 vol. 2, 2005. 7, 8

[7] Toni Buades, Yifei Lou, J. M. Morel, and Z. Tang. A note on multi-image denoising. *2009 International Workshop on Local and Non-Local Approximation in Image Processing*, pages 1–15, 2009. 2

[8] D. Butler, J. Wulff, G. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 6

[9] M. Chiang and T. Boult. Efficient super-resolution via image warping. *Image Vis. Comput.*, 18:761–771, 2000. 2

[10] Kostadin Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. *2007 15th European Signal Processing Conference*, pages 145–149, 2007. 2

[11] Kostadin Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16:2080–2095, 2007. 2, 7, 8

[12] Michel Deudon, A. Kalaitzis, Israel Goytom, M. R. Arefin, Zhichao Lin, K. Sankaran, Vincent Michalski, S. Kahou, Julien Cornebise, and Yoshua Bengio. Highres-net: Recursive fusion for multi-frame super-resolution of satellite imagery. *ArXiv*, abs/2002.06460, 2020. 1, 2, 7

[13] C. Dong, Chen Change Loy, Kaiming He, and X. Tang. Learning a deep convolutional network for image super-resolution. In *ECCV*, 2014. 4

[14] A. Dosovitskiy, P. Fischer, Eddy Ilg, Philip Häusser, Caner Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 6

[15] Michael Elad and A. Feuer. Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 6 12:1646–58, 1997. 1, 2

[16] E. Faramarzi, D. Rajan, and M. Christensen. Unified blind method for multi-image super-resolution and single/multi-image blur deconvolution. *IEEE Transactions on Image Processing*, 22:2101–2114, 2013. 2

[17] Sina Farsiu, Michael Elad, and P. Milanfar. Multiframe demosaicing and super-resolution from undersampled color images. In *IS&T/SPIE Electronic Imaging*, 2004. 1, 2, 3, 4

[18] Sina Farsiu, D. Robinson, Michael Elad, and P. Milanfar. Robust shift and add approach to superresolution. In *SPIE Optics + Photonics*, 2003. 2

[19] Sina Farsiu, M. Robinson, Michael Elad, and P. Milanfar. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing*, 13:1327–1344, 2004. 2

[20] C. Godard, K. Matzen, and Matthew Uyttendaele. Deep burst denoising. In *ECCV*, 2018. 2

[21] T. Gotoh and M. Okutomi. Direct super-resolution and registration using raw cfa images. In *CVPR 2004*, 2004. 2, 4

[22] R. Hardie, K. Barnard, John G. Bognar, E. Armstrong, and E. Watson. High-resolution image reconstruction from a sequence of rotated and translated frames and its application to an infrared imaging system. *Optical Engineering*, 37:247–260, 1998. 2, 3, 5

[23] M. Haris, Gregory Shakhnarovich, and N. Ukita. Recurrent back-projection network for video super-resolution. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3892–3901, 2019. 2

[24] S. W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, J. Barron, F. Kainz, Jiawen Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Transactions on Graphics (TOG)*, 35:1 – 12, 2016. 1, 2, 7, 8

[25] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[26] Y. He, Kim-Hui Yap, L. Chen, and Lap-Pui Chau. A nonlinear least square technique for simultaneous image registration and super-resolution. *IEEE Transactions on Image Processing*, 16:2830–2841, 2007. 2

[27] G. Healey and R. Kondepudy. Radiometric ccd camera calibration and noise estimation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16:267–276, 1994. 5, 7

[28] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. *arXiv preprint arXiv:2002.05509*, 2020. 6

[29] Eddy Ilg, N. Mayer, Tonmoy Saikia, Margret Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017. 3

[30] S. Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, abs/1502.03167, 2015. 6

[31] M. Irani and Shmuel Peleg. Improving resolution by image registration. *CVGIP Graph. Model. Image Process.*, 53:231–239, 1991. 2

[32] M. Kawulok, Pawel Benecki, Krzysztof Hrynczenko, D. Kostrzewa, S. Piechaczek, J. Nalepa, and B. Smolka. Deep learning for fast super-resolution reconstruction from multiple images. In *Defense + Commercial Sensing*, 2019. 1, 2

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 6, 8

[34] T. Köhler, X. Huang, Frank Schebesch, A. Aichert, A. Maier, and J. Hornegger. Robust multiframe super-resolution employing iteratively re-weighted minimization. *IEEE Transactions on Computational Imaging*, 2:42–58, 2016. 2

[35] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *ECCV*, 2018. 2

[36] Filippos Kokkinos and Stamatios Lefkimmiatis. Iterative residual cnns for burst photography applications. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5922–5931, 2019. 1, 2

[37] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, Serge Belongie, Victor Gomes, Abhinav Gupta, Chen Sun, Gal Chechik, David Cai, Zheyun Feng, Dhyanesh Narayanan, and Kevin Murphy. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017. 8

[38] Wei-Sheng Lai, Jia-Bin Huang, N. Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5835–5843, 2017. 4

[39] O. Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, J. Barron, Dillon Sharlet, Ryan Geiss, S. W. Hasinoff, Y. Pritch, and M. Levoy. Handheld mobile photography in very low light. *ACM Transactions on Graphics (TOG)*, 38:1 – 16, 2019. 1

[40] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140, 2017. 4

[41] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising using separable 4d nonlocal spatiotemporal transforms. In *Electronic Imaging*, 2011. 2

[42] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on Image Processing*, 21:3952–3966, 2012. 2, 7, 8

[43] Talmaj Marinc, V. Srinivasan, S. Gül, C. Hellge, and W. Samek. Multi-kernel prediction networks for denoising of burst images. *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2404–2408, 2019. 2, 6, 7, 8

[44] N. Mayer, Eddy Ilg, Philip Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4040–4048, 2016. 6

[45] Ben Mildenhall, J. Barron, Jiawen Chen, Dillon Sharlet, R. Ng, and Robert Carroll. Burst denoising with kernel prediction networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 2, 6, 7, 8

[46] Andrea Bordone Molini, Diego Valsesia, G. Fracastoro, and E. Magli. Deepsum: Deep neural network for super-resolution of unregistered multitemporal images. *IEEE Transactions on Geoscience and Remote Sensing*, 58:3644–3656, 2020. 1, 2

[47] Shmuel Peleg, D. Keren, and L. Schweitzer. Improving image resolution using subpixel motion. *Pattern Recognit. Lett.*, 5:223–226, 1987. 1, 2, 3

[48] L. Pickup, David P. Capel, S. Roberts, and Andrew Zisserman. Overcoming registration uncertainty in image super-resolution: Maximize or marginalize? *EURASIP Journal on Advances in Signal Processing*, 2007:1–14, 2007. 2

[49] L. Pickup, David P. Capel, S. Roberts, and Andrew Zisserman. Bayesian methods for image super-resolution. *Comput. J.*, 52:101–113, 2009. 2

[50] Edward T. Reehorst and Philip Schniter. Regularization by denoising: Clarifications and new interpretations. *IEEE Transactions on Computational Imaging*, 5:52–67, 2019. 2

[51] R. Schultz and R. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 5 6:996–1011, 1996. 2

[52] Jonathan R Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, Pittsburgh, PA, USA, 1994. 5

[53] W. Shi, J. Caballero, Ferenc Huszár, J. Totz, A. Aitken, R. Bishop, D. Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1874–1883, 2016. 6

[54] Deqing Sun, X. Yang, Ming-Yu Liu, and J. Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8934–8943, 2018. 3, 6

[55] H. Takeda, Sina Farsiu, and P. Milanfar. Robust kernel regression for restoration and reconstruction of images from sparse noisy data. *2006 International Conference on Image Processing*, pages 1257–1260, 2006. 2

[56] H. Takeda, Sina Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16:349–366, 2007. 2

[57] M. Tico. Multi-frame image denoising and stabilization. *2008 16th European Signal Processing Conference*, pages 1–4, 2008. 2

[58] Michael E. Tipping and Charles M. Bishop. Bayesian image super-resolution. In *NIPS*, 2002. 2

[59] A. S. Tripathi, Martin Danelljan, L. Gool, and R. Timofte. Few-shot classification by few-iteration meta-learning. In *ICRA*, 2021. 5

[60] R. Tsai and T. Huang. Multiframe image restoration and registration. In *Advance Computer Visual and Image Processing*, 1984. 2

[61] Singanallur V. Venkatakrishnan, C. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. *2013 IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013. 2

[62] Zhou Wang, A. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13:600–612, 2004. 7

[63] B. Wronski, Ignacio Garcia-Dorado, M. Ernst, D. Kelly, Michael Krainin, Chia-Kai Liang, M. Levoy, and P. Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38:1 – 18, 2019. 1, 2

[64] Zhihao Xia, Federico Perazzi, M. Gharbi, Kalyan Sunkavalli, and A. Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11841–11850, 2020. 1, 2, 6, 7, 8

[65] K. Zhang, L. Gool, and R. Timofte. Deep unfolding network for image super-resolution. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3214–3223, 2020. 2

[66] Kai Zhang, W. Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, 2017. 2

[67] Richard Zhang, Phillip Isola, Alexei A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018. 7

[68] Xuesong Zhang, J. Jiang, and S. Peng. Commutability of blur and affine warping in super-resolution with application to joint estimation of triple-coupled variables. *IEEE Transactions on Image Processing*, 21:1796–1808, 2012. 2

[69] Yulun Zhang, Yapeng Tian, Y. Kong, B. Zhong, and Yun Fu. Residual dense network for image super-resolution. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2472–2481, 2018. 4

[70] A. Zomet, A. Rav-Acha, and Shmuel Peleg. Robust super-resolution. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 1:I–I, 2001. 2