This CVPR Findings paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

Adaptive Continuous Kernel Networks for Image Reconstruction from Non-Uniform Sampling

Camille Biscarrat
MIT CSAIL

Michaël Gharbi
Reve

Rahul Goel
MIT CSAIL

Jonathan Ragan-Kelly
MIT CSAIL

Frédo Durand
MIT CSAIL

Tzu-Mao Li
UC San Diego

Figure 1. We propose a new neural network architecture for image reconstruction from non-uniform sampling patterns that properly account for the input samples' sub-pixel locations. We show applications to burst joint demosacking and denoising, and single image demosaicking and chromatic aberration correction (not shown here), where our model outperforms state-of-the-art baselines by a large margin. We show a single input frame of a real photograph from the HDR+ dataset [21], demosaicked for visualization (left), compared to previous joint demosaicking and denoising method from Wronski et al. [49], the Kernel Prediction Network [36] and Burstormer [14].

Abstract

Most deep learning image enhancement and reconstruction algorithms are restricted to images represented as square lattices of pixels. This becomes insufficient in several scenarios : 1) multiple frames with subpixel alignment, such as reconstructing an image from a noisy burst, 2) radial distortions, and 3) lateral chromatic aberration where the per wavelength warp leads to sample locations that are not only non-integer but different for the three color channels, violating the assumption of all demosaicking methods. We enable deep learning for image reconstruction with input samples at non-integer locations. We use subpixel sample location information and learn continuous reconstruction kernels, thereby maximally preserving information and avoiding degradation from resampling. The kernels are represented using neural networks conditioned on sample location as well as image information obtained from a coarser image reconstruction. Our model successfully demosaicks, denoises and merges stacks of burst frames across varying noise levels. We also demonstrate how this method can correct for chromatic aberrations in single images, making it, to our knowledge, the first joint denoising, demosaicking and chromatic aberration correction. Our project page is at https://people.csail.mit.edu/cjbisc/burst/.

1. Introduction

Neural networks have revolutionized image processing applications such as deblurring [2], demosaicking and denoising [18], and super-resolution [4]. However, most approaches are restricted to regular discrete grids of pixels that cannot deal with scenarios where input samples are at non-integer locations. This is unfortunately a common situation in computational photography, e.g., with burst processing on smartphones, radial distortion, or chromatic aberrations. When frames of a burst are aligned using a homography or affine transform, this results in arbitrary real-valued subpixel color sample locations. Single frame scenarios also lead to non-integer sample positions, such as when correcting radial distortion or lateral chromatic through warping or channel-dependent scaling (Figure 2).

Most existing methods either ignore these irregular sampling patterns, or resample input signals into a common

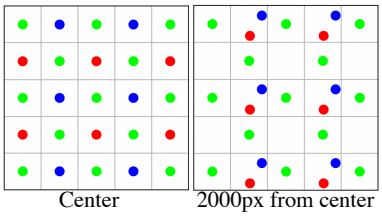

Figure 2. **Sampling pattern of a real image with lateral chromatic aberration.** The distribution of color samples varies across the image. At the center of the image, all samples are on the pixel grid. As we move away from the center, red and blue samples no longer lie on the grid.

grid, leading to a loss of signal and resolution. For chromatic aberrations correction, traditional approaches first perform demosaicking and later apply a per-channel spatial scale. In addition to the drawbacks of resampling, this means that the demosaicking step uses poorly-aligned channels, which violates the assumption of most such methods. Instead, we preserve the full geometric accuracy of our input sample geometry and design a deep learning approach to image reconstruction that can handle non-integer sample locations.

Rather than directly learning the output image, we learn continuous reconstruction kernels, represented using per-pixel fully-connected networks. At each output pixel, we evaluate its continuous kernel at all sample locations within a fixed radius, and combine the weighted samples to obtain the output pixel value (Figure 3). The weighted sum thus depends on the local image features as well as the sampling pattern. The kernel shape is parametrized using a spatially-varying latent code, produced by an encoder jointly trained with the kernel network. In contrast to treating the samples as completely irregular point cloud and processing them as such [23, 41, 47], our design allows us to apply state-of-the-art pixel-based architectures [53] as the encoder, while accounting for the subpixel locations of the samples.

We evaluate our approach on burst joint demosaicking and denoising, and we further show that our method can handle single image chromatic aberration correction from a single raw image. To our knowledge, this is the first deep learning solution for joint demosaicking, denoising, and chromatic abberation correction. We show that our learned continuous kernels lead to clean output images and better reconstruct previous work that do not account for irregular sampling patterns.

2. Related work

Burst processing of raw frames. Computational photography methods have long used the information from multiple raw frames to obtain a higher quality composite image [20, 22, 45], for applications like denoising [13, 36, 50], demosaicking and super-resolution [6, 10, 14, 15, 27, 33, 34, 46],

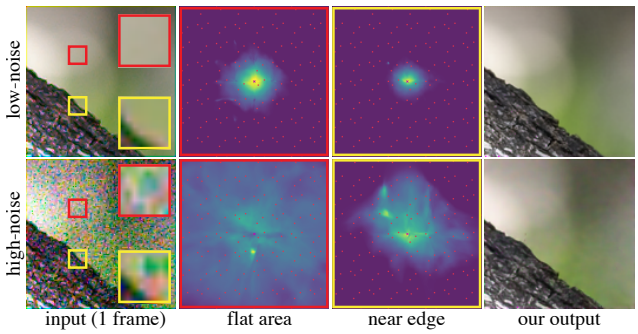

Figure 3. **Continuous kernels adapt to the sub-pixel image structure.** Our learned per-pixel continuous kernels (columns 2 and 3) can precisely filter samples with sub-pixel locations while adapting to the local image content. For instance, the kernel support tends to be small when the noise level is low (top), but grows larger as noise increases (bottom). The kernel shapes also adapt to local image structures, like corners (yellow inset), but remain anisotropic in flat image regions (red inset). Red dots show the local sampling pattern.

HDR imaging [28] or deblurring [2]. Hasinoff et al. [21] developed an end-to-end pipeline that aligns and fuses raw frames before demosaicing and post-processing the final image for HDR and low-light conditions. Liba et al. [30] focuses on handheld mobile photography in very low light. Closest to our work, Wronski et al. [49] pushed the idea of exploiting camera shake as an additional source of information by treating the pixels in a burst collectively as samples of a continuous signal. Samples are merged using a family of continuous Gaussian kernels. The kernels adapt to the local scene and better preserves detail while performing denoising and demosaicing. However, the kernels are limited to Gaussians, which can be sub-optimal. By learning continuous kernels, instead of restricting their shape to a simple parametric form, we can adapt to sparse local information more precisely and produce more accurate reconstructions of the continuous scene.

Image processing with deep learning. Deep learning has proven itself successful at various image processing tasks, such as deblurring [2], joint demosaicing and denoising [18] and super-resolution [4]. In the burst processing context, Mildenhall et al. and Xia et al. [36, 50] use convolutional neural networks to produce discrete kernels for burst image reconstruction. They learn per-pixel reconstruction kernels that merge frames into a denoised output. Because the kernels are discrete, the resolution of the output reconstruction is limited by the kernel resolution. Bhat et al. [5] instead learn the image degradation model and prior and solve an inverse problem to merge burst images, but still rely on networks that process discrete pixels. We avoid the loss of resolution by learning continuous kernels that can be evaluated at arbitrary sub-pixel locations. Recent methods, use a variety of

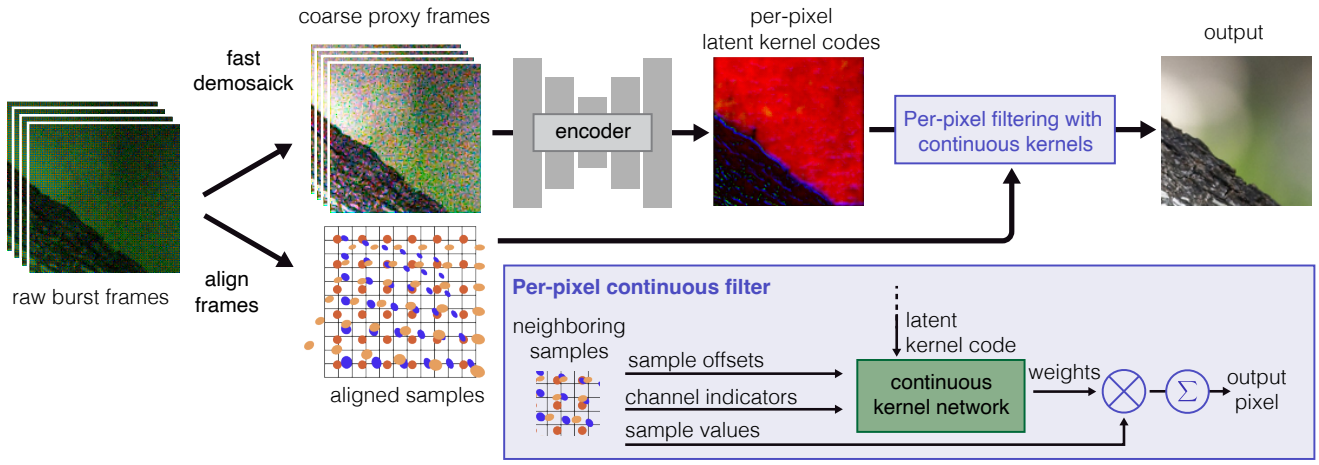

Figure 4. **Overview.** Our approach reconstructs a clean, demosaicked image starting from a noisy burst of misaligned raw frames, or misaligned channels from the same image in the case of chromatic aberrations **(main figure)** . The raw frames are converted to RGB using a naive, fast demosaicking algorithm before the encoding stage. The encoder outputs a latent vector per pixel, which modulates the shape of the continuous kernel for this pixel. The kernel function is a trained fully-connected network that operates on the samples in a fixed spatial window around the pixel. This network **(inset)** takes as input the samples' offsets from the pixel center, a one-hot encoding of the color channel, and the latent code, and outputs a kernel weight. The final pixel value is obtained by summing the weighted sample contributions in the neighborhood. Both encoder and kernel networks are trained end-to-end.

architectures and designs, such as U-Nets [42], Deformable convolutions [11], and more recently transformer-based or hybrid convolution-transformer methods, like SwinTransformer [31], SwinIR [29], Restormer [52], Uformer [48] or SCUNet [53] (which our encoder backbone uses). Kernel-predicting convolutional networks have been used to denoise images [36] and Monte Carlo renderings [3], down to the sub-pixel sample level [19].

Learning Continuous Functions. Implicit neural representations have gained in popularity for volumetric multi-view synthesis [37, 44] and other image reconstruction [8] or generation [9] tasks. See Xie et al. [51]'s article for a survey. These approaches train neural networks, typically MLPs rather than convolutional networks, to recover a continuous scene representation (3D or 2D) from sparse samples. This continuous signal is then sampled to obtain an image. These methods can recover higher frequency information using Fourier encodings with ReLU [37] or sinusoidal activations [43], resulting in more detailed outputs. The approaches typically fit a large network to a single scene or image. Recent approaches introduced local models that can be conditioned on an input signal [35] to bypass the costly run-time optimization of earlier techniques. We use neural implicits to model reconstruction kernels, instead of the signal directly, which generalizes beyond the training set. Neural radiance fields have been used for burst denoising [38, 39], but these approaches require expensive volume rendering at runtime and are typically targeted for multi-view inputs with a larger baselines than typical bursts.

3. Method

Our goal is to reconstruct an image from a collection of sparse noisy input samples with a non-uniform spatial distribution. In typical applications, these samples are the color measurements of a single raw image or a burst of such images captured in quick succession. The samples are usually located at arbitrary non-integer locations relative to the output pixel grid, *e.g.* because of misalignments between the burst frames or between the color channels of each image due to optical aberrations.

We want to preserve the full geometric precision of the sample locations (§ 3.1) to resolve finer details while exploiting deep learning's ability to analyze complex image pattern and learn strong priors. This requires a new network design. Our approach is illustrated in Fig. 4. Our model predicts, for each output location, a *continuous kernel* whose shape adapts both to the local image structure and to the local sampling pattern (§ 3.2). To adapt the kernels to the image content, we start with a regular grid-based neural network that analyzes the stack of discrete input frames at their original resolution and encodes this information in a latent feature map z (§ 3.3). At each pixel, the latent code conditions a fully-connected network, thus defining a continuous kernel that assigns a scalar weight to arbitrary locations around the output pixel. The output is the weighted sum of all samples in a fixed-size neighborhood of the pixel.

In the rest of the section, we consider the task of burst denoising and demosaicking to better ground the exposition. Section 5 discusses another application to chromatic

aberration correction for single images.

3.1. Reconstruction from non-uniform samples

Our key idea is to consider input raw images as a set of samples with non-uniform continuous coordinates rather than a regular discrete grid of pixels. For burst imaging, these samples come from the burst frames after alignment onto a common reference coordinate system (that of the output). Each sample, indexed by $k \in \mathbb{N}$, is a tuple $(\mathbf{x}_k, f_k, v_k)$, where $\mathbf{x}_k \in \mathbb{R}^2$ is the sample's location in the reference frame, $f_k \in \mathbb{R}$ is the sample's scalar value, and $v_k \in \{0,1\}^3$ is a one-hot vector specifying the color channel of the current sample, *e.g.*, $[1,0,0]$ for red. Samples are typically noisy. We discuss our noise model in Section 3.4.

Alignment. We obtain alignment by computing the homographies between the frames and the output grid using SIFT features [32] matching with RANSAC [16], though any method that yields subpixel alignment can be used, such as block matching and optical flow as in Wronski et al. [49].

Image reconstruction. We reconstruct the final image I as a weighted average of samples in a local neighborhood around each output pixel. Specifically, the pixel value at pixel $p \in \mathbb{N}$ and color channel $c \in \{0,1,2\}$ is given by:

$$I_{pc} = \frac{\sum\limits_{k \in \mathcal{K}_p} f_k \cdot w_{kc}}{\sum\limits_{k \in \mathcal{K}_p} |w_{kc}| + \epsilon}, \tag{1}$$

where $w_{kc} \in \mathbb{R}$ is the (possibly negative) sample weight extracted from the reconstruction kernel, and $\epsilon = 1e^{-7}$. The set $\mathcal{K}_p$ contains all samples whose distance to pixel p is under a fixed kernel radius, which we set to 4 (in output pixel units).

Collectively, the sub-pixel samples provide fine information about the underlying image structure [49]. This lets us resolve finer details than previous approaches that resample the burst frames [36]. To maximize cross-channel information sharing, we let every sample, regardless of color, contribute to all 3 output channels. The weights w_{kc} are obtained by sampling a pixel-specific continuous kernel centered at pixel p, which we describe next.

3.2. Per-pixel continuous kernel network

We parametrize the per-pixel continuous kernel at p for color channel c using a fully-connected neural network G_{pc}. These fully-connected networks allows us to query at arbitrary spatial locations, unlike convolutional models that are restricted to regular integer grid. Our network has 8 hidden layers of dimension 128 with Leaky ReLU activations. We omit the activation in the last layer to allow positive and negative output weights, since we want image filter weights to be negative if needed, *e.g.*, to serve as sharpening filters.

The weight for sample $k \in \mathcal{K}_p$, is then given by

$$w_{kc} = G_{pc}(\delta_{kp}, v_k, z_{pc}), \tag{2}$$

$\delta_{kp} \in [-1,1]^2$ is the signed spatial offset between sample k and output pixel p, normalized by the kernel radius. The one-hot vector v_k is the sample channel indicator and allows the network to adapt to the sample color, *e.g.*, weighting blue samples more heavily than red ones when reconstructing the blue channel. Finally, $z_{pc} \in \mathbb{R}^d$ is a learned latent vector that modules the kernel shape according to local image information, so that each pixel can have a different kernel. Figure 3 shows a few kernels, adapting to the local image content and sampling pattern.

Next, we describe how our model estimates the latent kernel codes from the input frames.

3.3. Latent kernel code computation

To obtain the latent codes, we train an encoder network jointly with the kernel networks. Unlike our filtering stage, this encoder operates on regular pixel grids and relies on convolutional layers to leverage local image information and spatial interactions between neighboring pixels. The image frames are first demosaicked using a simple and fast green-based demosaicking algorithm [24], then stacked and passed through the encoder network (without any alignment or resampling). The output is a tensor containing the latent vectors for the three output channels at each pixel. We use a modified SCUNet [53] backbone as our encoder, a UNet architecture that uses Swin-Convolution Transformer blocks to better capture data-dependent interactions between the pixels. It takes as input a stack of N burst frames, i.e. $3N$ channels, and outputs a tensor with depth $3|z_{pc}|$ of latent codes, one per output pixel per channel.

Thanks to the per-pixel latent codes, our model can adapt the continuous kernels to local scene information (Fig. 3). In smooth image regions, larger kernels average more samples to better suppress image noise. Near image edges, the kernel are more discriminative, assigning more weight to samples on the same side of the edge and avoiding crossing the edge, leading to sharper output features. Up to the max neighborhood size, the kernel support depends on the noise level in the samples: noisier inputs require more samples to be combined for stronger denoising. The encoder network also uses inter-frame information to adapt the kernel shape to the burst sampling pattern; see the appendix for details.

3.4. Training Data

Because large scale datasets of image bursts and reconstruction ground truth pairs are not readily available, we train our model on a synthetic dataset, obtained by creating burst of raw frames from individual sRGB images. Specifically, we use the dataset introduced by Gharbi et al. [18] to focus

our training on image patterns that are difficult to demosaick. Starting from a gamma corrected image, we unprocess it into a pseudo-linear RGB following Brooks et al. [7]. This is our ground truth. We then sample $N = 5$ random translation offsets from a 2D normal distribution with variance 2 to generate misaligned frames, mimicking the natural hand tremor in burst photography [49]. We simulate sensor noise as additive signal-dependent Gaussian noise [17], $y \sim \mathcal{N}(x, \sigma_r^2 + \sigma_s x)$, where y is the noisy measurement of the true pixel intensity x. The read and shot noise parameter, (σ_r, σ_s), are randomly sampled from a realistic distribution as described by Brooks et al. [7]. They are fixed for a given burst, but vary throughout the dataset. Finally, we simulate the mosaicked signal for each frame of the burst by discarding 2 out of 3 color channels at each pixel according to the Bayer pattern.

Working with images in the raw linear domain allows us to better emulate a real-world image processing pipeline, avoiding the loss of signal information and artifacts introduced by post-processing steps.

Our method requires the translation offsets between burst frames to align them onto the same reference axis. During training with our synthetic dataset, the exact translations are known, but we nonetheless simulate small random error to increase our model's robustness at test time to alignment calculation errors for real bursts with unknown shifts. For this, we add a jitter sampled from a uniform distribution in $[-0.25, 0.25]$ to the translations.

3.5. Training Details

We train our model using an L_2 loss on the pixel values: $\mathcal{L}(I, I_{GT}) = \|I - I_{GT}\|_2^2$, where I is the output and I_{GT} the corresponding ground truth image. We optimize the encoder and kernel network using the ADAM optimizer [12] with a learning rate of 10^{-4} and decrease the learning rate by a factor of 2 every 15 epochs. We train on bursts of 5 frames, with 64×64 resolution and a batch size of 4. Our sample neighborhoods $\mathcal{K}$ have size $K = 320$ in a 8×8px window and the latent have dimension $|z_{pc}| = 8$. We initialize our encoder with the pretrained SCUNet weights, but found that training also converge to the same solution when starting from random initial weights.

3.6. Efficient Inference

Because the sample locations are non-integer, grid-based methods cannot be used to obtain the local neighborhood of each output pixel. However, for each output pixel, input samples remain spatially restricted to their local neighborhood. We efficiently search samples for each output pixel by clustering the samples into buckets using K-Means. Then for an output pixel query, we find the closest few buckets and perform an exhaustive search over the samples to find the K closest samples to the output coordinates. We develop an efficient GPU implementation for inference using the FAISS library [25].

After obtaining the K closest samples, their weights are calculated using the three 8-layer per-pixel continuous kernel MLPs. As image size increases, this becomes expensive to compute. We observe that, for each output pixel, some subset of samples are more important than others for the reconstruction. To this end, we first compute the output pixel value with a small set of nearby samples and update the result as we gradually grow this set towards K. We stop growing the neighborhood $\mathcal{K}$ when the updated output value minimally differs from the previously calculated value. We observe no degradation in visual quality with this subset sampling scheme and report PSNR and SSIM values for various noise levels in Tables 1, 2.

For a 512×512 burst of size $N = 5$ with kernel size 8, our implementation reconstructs the image in 17s on an older NVIDIA Titan Xp and in 2.2s on a modern NVIDIA H200 GPU.

4. Burst Demosaicking and Denoising

We use our method to learn continuous kernels to jointly demosaick and denoise bursts of images captured in quick succession. After alignment the burst can be viewed as a noisy, irregular sub-pixel sampling of the scene photographed. We evaluate our method on both synthetic and real burst, comparing against several baselines.

4.1. Evaluation on synthetic data

For all experiments in this section, we use a synthetic test set of 128×128 bursts with N=5 frames and 4 different noise levels. The test set is generated according to Section 3.4 using 100 sRGB images from the DIV2K dataset [1]. We compare to the following baselines:

- "Burstormer" [14] color model works on demosaicked, gamma un-compressed RGB inputs and takes an estimate of the image noise as an additional input.
- Kernel-Predicting Networks "KPN" [36], which has been modified to learn per-color kernels for 3 channel, demosaiced input images since the original method works on grayscale images only.
- "KPN+" replaces KPN's UNet backbone with a SCUNet [53] with attention layers, for direct comparison with our method.
- "Wronski et al. [49]" uses locally varying continuous Gaussian kernels for reconstruction.
- "Demo+" demosaics each burst frame using the Demosaicnet model from Gharbi et al. [18] then aligns and averages the frames to produce the output.

Both KPN baselines are trained on our data. For SCUNet, we use the pretrained model without finetuning. We compute PSNR and SSIM for all images after sRGB gamma encoding, since this space is more per-

Method	Gain $\propto 1$		Gain $\propto 2$		Gain $\propto 4$		Gain $\propto 8$	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Wronski	28.49	0.733	25.94	0.599	22.26	0.411	18.80	0.268
Demo+	34.57	0.912	31.99	0.842	28.17	0.685	24.46	0.508
KPN	33.28	0.876	30.54	0.793	26.92	0.652	22.27	0.423
KPN+	34.12	0.889	31.16	0.806	27.69	0.675	23.76	0.479
Burst.	34.27	0.886	31.99	0.823	29.46	0.765	**26.72**	**0.702**
Ours	**36.46**	**0.932**	**33.54**	**0.875**	**30.12**	**0.779**	26.45	0.623
w/ SS	36.46	0.932	33.54	0.875	30.10	0.778	26.42	0.622

Table 1. **Demosaicking and denoising vs. noise level.** We compare our method to 6 baselines on a synthetic test set of 100 5-frames bursts, derived from the DIV2K dataset [1], using the same noise and motion models as our training data. "w/ SS" is our model with the subset sampling procedure detailed in Section 3.6. Our network was not trained on noise levels of the rightmost column.

ceptually meaningful than linear values. Table 1 summarizes our result for noise levels corresponding to sensor gains $(1, 2, 4, 8)$. These correspond to read and shot noise values $(\log \sigma_r, \log \sigma_s) \rightarrow (-2.2, -2.6), (-1.8. -2.2), (-1.4, -1.8), (-1.1, -1.5)$. Our model is not trained on the highest noise level. Figure 5 shows examples from our synthetic test set. Our method successfully denoises while retaining more details. Our method quantitatively and qualitatively outperforms the baselines at most noise levels.

4.2. Evaluation on real bursts

We show qualitative comparisons to a production algorithm (HDR+ [21]) on real raw bursts of 512×512 in Figures 1 and 6. We provide quantitative comparisons using the HDR+ processed output as the ground truth in Table 2. The latter involves a more extensive post-processing pipeline. To account for the resulting mismatch in color and other misalignments between the output frame and the reference, we align then color-correct our outputs to match the ground truth before calculating the PSNR, as described in [4]. Our approach again shows superior denoising performance and better detail preservation, despite using only 5 frames (the reference provided in this dataset uses 10). Additional results can be found in the appendix.

4.3. Ablations

In Table 3, we show 3 ablations to illustrate the benefit of our continuous kernels and the importance of sub-pixel samples. We report PSNR for shot noise $\sigma_s = 0.005$. "Discrete kernels" replaces our MLP-based continuous kernel with a discrete 9×9 kernel with identical support. We bilinearly interpolate the discrete kernel at sub-pixel sample locations prior to filtering. Here, the encoder predicts the 9×9 kernel values for each pixel rather than latents. "Direct RGB" does away with the kernels and predicts the RGB values directly with the encoder. "Binned δ_{kp}" quantizes the sub-pixel sample offsets to the nearest output pixel coordinate before passing them as input to the kernel network.

Method	PSNR (↑)
Wronski et al. [49]	27.68
Demo+	28.52
KPN [36]	28.18
KPN+	28.08
Burstormer	28.05
Ours	**28.89**
w/ SS	28.87

Table 2. **Burst demosaicking and denoising on real images.** We evaluate our method on a real test set of 70 5-frame bursts from the HDR+ dataset. "w/ SS" uses our model with the subset sampling procedure detailed in Section 3.6. We use the AlignedPSNR metric [4] to correct the mismatch in alignment and post-processing between the bursts and ground truth.

Discrete Kernels	Direct RGB	Binned δ_{kp}	Ours
30.55	31.22	33.04	**33.96**

Table 3. **Ablations.** PSNR over our synthetic test set for different ablations of our model. See Section 4.3 for the description.

m	0	0.01	0.05	0.1	0.25	0.5	1
Translation	36.46	36.46	36.43	36.35	35.83	34.43	31.83
Full matrix	36.46	36.44	35.82	34.47	31.30	28.65	26.36

Table 4. **Alignment noise.** PSNR over our synthetic test for different levels of alignment noise. For "Translation", we add noise to the translation terms only; "Full matrix" adds noise to the scale and shear terms as well. See Section 4.3 for more details.

To determine our model's sensitivity to perfect alignment, we evaluate our test set with increasing noise on the known alignment matrices. The noise is randomly sampled from a normal distribution with variance $m\sigma$, where $\sigma_{tr} = 1, \sigma_{sc} = 0.005, \sigma_{sh} = 0.0005$ for the translation, scale and shear coefficients, respectively, and m is a scalar. Table 4 shows the PSNR for different values of m. In the first row, we add noise to the translation offsets only. In the second row, we also add noise to the scale and shear coefficients. We fix the image noise level of the bursts to $(\log(\sigma_s), \log(\sigma_r)) \rightarrow (-2.2, -2.6)$. Our model corrects for small translation misalignments, within the range seen in our training data. We can make our model more robust to small alignment errors by expanding our training set to include noise on the shear and scale terms as well.

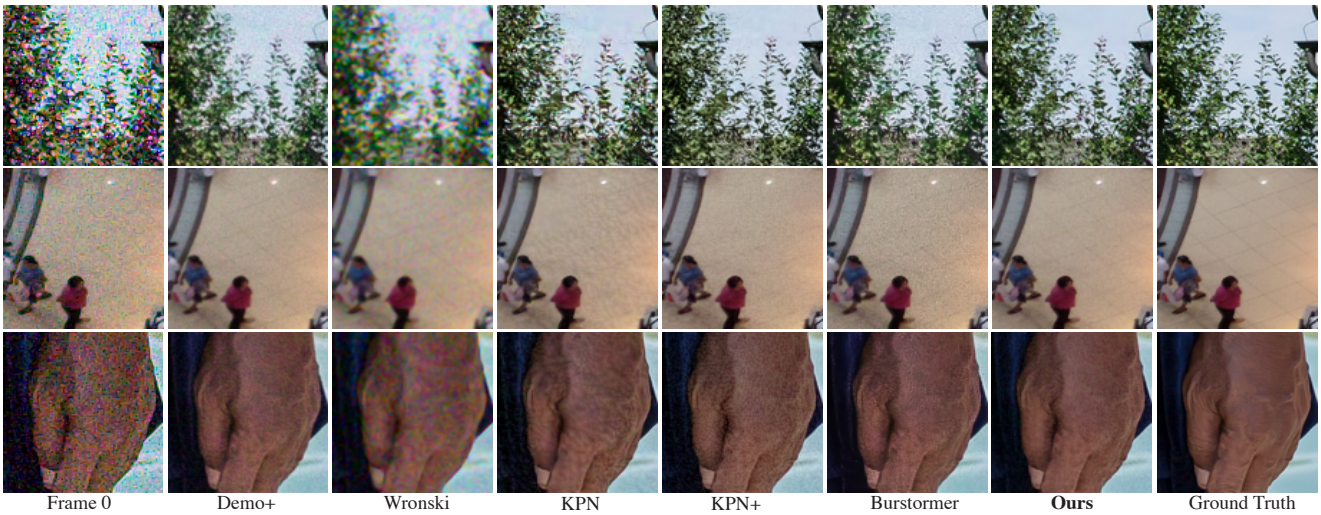

Figure 5. **Burst demosaicking and denoising on synthetic data.** Our model (second to last) consistently outperforms previous works and baselines (columns 2–6) on our synthetic 5-frames burst demosaicking and denoising test set. One input frame is shown demosaicked for visualization (left), as well as the ground truth (right).

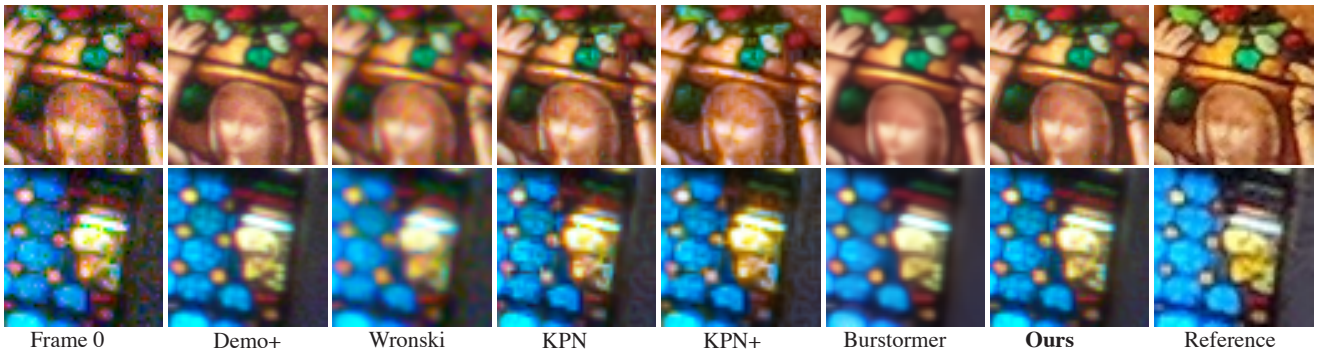

Figure 6. **Demosaicking and denoising real bursts.** Once trained, our model generalizes to real images. We show demosaicking and denoising results on 5-frames bursts from the HDR+ dataset [21]. The HDR+ result using 10 frames is shown for reference. It applies different post-processing after demosaicking and denoising, so tone and geometry are expected to be different from the inputs and our outputs. The input is demosaicked for visualization. Our outputs better preserve fine color details, like facial features and leaves (top row) or the details in the blue tile and orange dots (bottom row). Results best appreciated zooming in.

5. Chromatic Aberrations Correction

We demonstrate our approach generalizes to other imaging applications with irregular sampling patterns with the task of jointly demosaicking, denoising and correcting chromatic lens aberrations [26]. Chromatic aberrations (CA) are due to variations of the index of refraction, resulting in different lens power for different wavelengths. Lateral chromatic aberrations (LCA) mostly manifest as the RGB channels having different magnifications. In real images, such small magnification differences can result in 2-3px difference at 2000px from the optical center. We model LCA as a uniform scaling from the center of the image of the red and blue channels relative to the green one.

In this case, by analogy with the burst imaging task, the set of "frames" are the three channels of a single image. After alignment, red and blue samples are at sub-pixel locations with respect to the green sampling grid (Figure 2). We generate a synthetic training dataset for this task similarly to Section 3.4: instead of sampling random translation offsets, we randomly sample magnification factors for the red and blue channels from a uniform distribution such that the maximum color shift is of 4px at 2000px from the center of the image. We rescale the red and blue channels according to these factors to simulate LCA. For real test images, the scaling factors are calculated using SIFT and RANSAC applied to the three channels viewed as greyscale images.

Results on synthetic data. For all experiments in this section, we use a synthetic test set of 128×128 patches and 6 different noise levels. The test set is generated similarly to

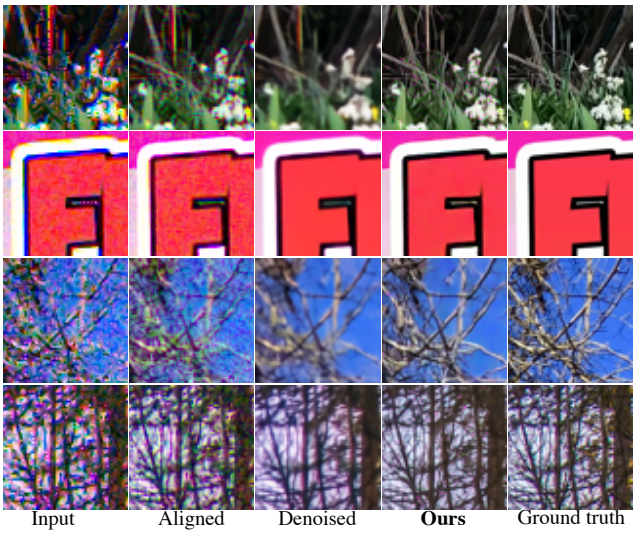

Figure 7. **Chromatic aberrations correction.** On our synthetic test set, our method outperforms the baseline in correcting lateral chromatic lens aberrations and removing noise while preserving more details. The input is shown demosaicked for visualization before alignment (first column), and after (second) for reference. Both our method and the Denoised baseline use the same ground-truth per-channel warping field for alignment.

Method	PSNR (↑)	SSIM (↑)
Demosaick, Align & Denoise	30.91	0.849
Ours	**33.41**	**0.877**

Table 5. **Chromatic aberration correction.** Our method significantly outperforms the standard approach for chromatic aberration correction: demosaick the image, then warp the channels into alignment given a geometric lens distortion model. Results are averaged over all noise levels of our DIV2K syntehtic test set.

the training set, using 100 sRGB images from the DIV2K dataset [1]. Commercial imaging software typically deals with these artifacts using a naive two-stage procedure: first demosaick the image, then warp the channels into alignment given a geometric lens distortion model [40]. This approach is problematic because demosaicking often creates false colors in the presence of noise and LCA, as it wrongly assumes the color channels are aligned. These artifacts are then smeared by the warping operation and become particularly salient at higher noise levels. We define a baseline sequential approach similarly: demosaick the image with [18], align the channels using the computed scale factors then denoise with SCUNet [53]. Our continuous kernels directly account for the shifted samples positions, with sub-pixel precision, leading to better reconstruction and robustness to noise, as shown in Figure 7 and Table 5.

Results on real images. We show results on real raw images in Figure 8. While our simplistic LCA model succesfully reduces color artifacts, it does not accurately model all

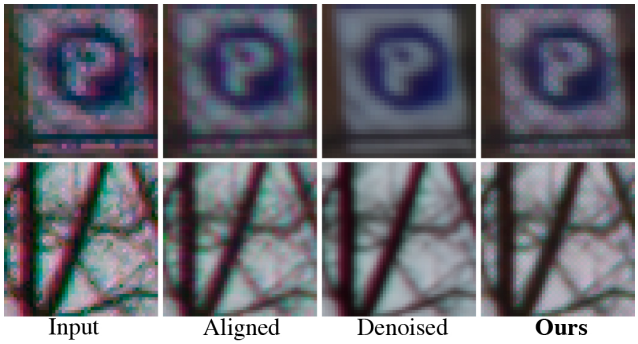

Figure 8. **Joint demosaicking, denoising and chromatic aberration correction on real images.** Our method better preserves fine details, like the chain link fence cross pattern (top row) and the thinner branches (bottom row). Results are best appreciated fully zoomed in. The input is demosaicked for visualization.

distortions due to the lens system. When the lens model is known, we expect a manual calibration step to lead to more accurate channel alignment and thus better reconstructions. The appendix contains additional results.

6. Limitations and Future Work

Our method has a few limitations. Our encoder still operates on naive rasterization of the input frames. Although the runtime cost may become prohibitive, it would be interesting to explore if further quality gains can be obtained by working with the raw samples at this stage as well. A more informed selection criteria for the subset sampling scheme could also improve reconstruction quality and performance. Another improvement could look into more complex parametrization of the kernels, using positional encoding or sine activations for higher frequency responses [43, 44]. Our continuous kernels would lend themselves naturally to resampling of the output at arbitrary (and continuously variable resolutions), e.g. for joint demosaicking/denoising/super-resolution. This is an interesting avenue for future work, which we have yet to experiment with.

7. Conclusions

We have presented a new deep learning architecture for image restoration from irregular sampling patterns of a scene. Our method uses an encoder network to analyse the scene structure and predict a latent code per output pixel. In turn, the latent codes parametrize a fully-connected network to represent spatially-varying per-pixel continuous kernels. These kernels are evaluated at sub-pixel sample locations to compute the final output pixel values as a weighted combination of input samples. Our evaluation shows our approach leads to significant image quality improvements on two common image processing tasks: burst joint demosaicking and denoising, and demosaicking, denoising and chromatic aberration correction for single images.

Acknowledgments
This work was supported in part by MIT GIST and MIT MGAIC.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2017. 5, 6, 8

[2] Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 731–747, 2018. 1, 2

[3] Steve Bako, Thijs Vogels, Brian McWilliams, Mark Meyer, Jan Novák, Alex Harvill, Pradeep Sen, Tony Derose, and Fabrice Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph.*, 36(4):97–1, 2017. 3

[4] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deep burst super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9209–9218, 2021. 1, 2, 6

[5] Goutam Bhat, Martin Danelljan, Fisher Yu, Luc Van Gool, and Radu Timofte. Deep reparametrization of multi-frame super-resolution and denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2460–2470, 2021. 2

[6] Goutam Bhat, Michaël Gharbi, Jiawen Chen, Luc Van Gool, and Zhihao Xia. Self-supervised burst super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10605–10614, 2023. 2

[7] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11028–11037, 2019. 5

[8] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8628–8638, 2021. 3

[9] Yinbo Chen, Oliver Wang, Richard Zhang, Eli Shechtman, Xiaolong Wang, and Michael Gharbi. Image neural field diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8007–8017, 2024. 3

[10] Wooyeong Cho, Sanghyeok Son, and Dae-Shik Kim. Weighted multi-kernel prediction network for burst image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 404–413, 2021. 2

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[12] P Kingma Diederik. Adam: A method for stochastic optimization. *(No Title)*, 2014. 5

[13] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Burst image restoration and enhancement. In *Proceedings of the ieee/cvf Conference on Computer Vision and Pattern Recognition*, pages 5759–5768, 2022. 2

[14] Akshay Dudhane, Syed Waqas Zamir, Salman Khan, Fahad Shahbaz Khan, and Ming Hsuan Yang. Burstormer: Burst image restoration and enhancement transformer. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2023. 1, 2, 5

[15] Sina Farsiu, Michael Elad, and Peyman Milanfar. Multiframe demosaicing and super-resolution of color images. *IEEE Transactions on Image Processing*, 15:141–159, 2006. 2

[16] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 4

[17] Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian. Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17:1737–1754, 2008. 5

[18] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 1, 2, 4, 5, 8

[19] Michaël Gharbi, Tzu-Mao Li, Miika Aittala, Jaakko Lehtinen, and Frédo Durand. Sample-based monte carlo denoising using a kernel-splatting network. *ACM Transactions on Graphics (TOG)*, 38(4):1–12, 2019. 3

[20] Tomomasa Gotoh and Masatoshi Okutomi. Direct super-resolution and registration using raw cfa images. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages II–II. IEEE, 2004. 2

[21] 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 (ToG)*, 35(6):1–12, 2016. 1, 2, 6, 7

[22] 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):1–13, 2014. 2

[23] Pedro Hermosilla, Tobias Ritschel, Pere-Pau Vázquez, Àlvar Vinacua, and Timo Ropinski. Monte Carlo convolution for learning on non-uniformly sampled point clouds. *ACM Transactions On Graphics (TOG)*, 37(6):1–12, 2018. 2

[24] Keigo Hirakawa and Thomas W. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IEEE International Conference on Image Processing*, 3:669–672, 2003. 4

[25] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 5

[26] Jan Tore Korneliussen and Keigo Hirakawa. Camera processing with chromatic aberration. *IEEE Transactions on Image Processing*, 23(10):4539–4552, 2014. 7

[27] Bruno Lecouat, Jean Ponce, and Julien Mairal. Lucas-Kanade reloaded: End-to-end super-resolution from raw image bursts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2370–2379, 2021. 2

[28] Bruno Lecouat, Thomas Eboli, Jean Ponce, and Julien Mairal. High dynamic range and super-resolution from raw image bursts. *ACM Trans. Graph.*, 41(4), 2022. 2

[29] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 3

[30] Orly Liba, Kiran Murthy, Yun-Ta Tsai, Tim Brooks, Tianfan Xue, Nikhil Karnad, Qiurui He, Jonathan T Barron, Dillon Sharlet, Ryan Geiss, et al. Handheld mobile photography in very low light. *ACM Transactions on Graphics (TOG)*, 38 (6):1–16, 2019. 2

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[32] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 4

[33] Ziwei Luo, Lei Yu, Xuan Mo, Youwei Li, Lanpeng Jia, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Ebsr: Feature enhanced burst super-resolution with deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 471–478, 2021. 2

[34] Ziwei Luo, Youwei Li, Shen Cheng, Lei Yu, Qi Wu, Zhihong Wen, Haoqiang Fan, Jian Sun, and Shuaicheng Liu. Bsrt: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 998–1008, 2022. 2

[35] Ishit Mehta, Michaël Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. *arXiv preprint arXiv:2104.03960*, 2021. 3

[36] 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. 1, 2, 3, 4, 5, 6

[37] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020. 3

[38] Ben Mildenhall, Peter Hedman, Ricardo Martin-Brualla, Pratul P Srinivasan, and Jonathan T Barron. NeRF in the dark: High dynamic range view synthesis from noisy raw images. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16190–16199, 2022. 3

[39] Naama Pearl, Tali Treibitz, and Simon Korman. Nan: Noise-aware nerfs for burst-denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12672–12681, 2022. 3

[40] Stefan Petersson, Håkan Grahn, and Jim Rasmusson. Blind correction of lateral chromatic aberration in raw bayer data. *IEEE Access*, 9:99455–99466, 2021. 8

[41] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 2

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 3

[43] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020. 3, 8

[44] Matthew Tancik, Pratul P Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *arXiv preprint arXiv:2006.10739*, 2020. 3, 8

[45] Roger Y. Tsai and Thomas S. Huang. Multiframe image restoration and registration. 1984. 2

[46] Patrick Vandewalle, Karim Krichane, David Alleysson, and Sabine Süsstrunk. Joint demosaicing and super-resolution imaging from a set of unregistered aliased images. In *Digital Photography III*, pages 78–89. SPIE, 2007. 2

[47] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph CNN for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019. 2

[48] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022. 3

[49] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38(4): 1–18, 2019. 1, 2, 4, 5, 6

[50] Zhihao Xia, Federico Perazzi, Michaël Gharbi, Kalyan Sunkavalli, and Ayan Chakrabarti. Basis prediction networks for effective burst denoising with large kernels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11844–11853, 2020. 2

[51] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. In *Computer Graphics Forum*, pages 641–676. Wiley Online Library, 2022. 3

[52] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 3

[53] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via swin-conv-unet and data synthesis. *Machine Intelligence Research*, 20:822–836, 2023. 2, 3, 4, 5, 8