# Robust Image Denoising using Kernel Predicting Networks

Zhilin Cai[2] Yang Zhang[1] Marco Manzi[1] Cengiz Oztireli[1] Markus Gross[1,2] Tunç Ozan Aydın[1]

DisneyResearch|Studios[1] ETH Zurich[2]

**Abstract**

*We present a new method for designing high quality denoisers that are robust to varying noise characteristics of input images. Instead of taking a conventional blind denoising approach or relying on explicit noise parameter estimation networks as well as invertible camera imaging pipeline models, we propose a two-stage model that first processes an input image with a small set of specialized denoisers, and then passes the resulting intermediate denoised images to a kernel predicting network that estimates per-pixel denoising kernels. We demonstrate that our approach achieves robustness to noise parameters at a level that exceeds comparable blind denoisers, while also coming close to state-of-the-art denoising quality for camera sensor noise.*

**CCS Concepts**

• ***Computing methodologies*** → ***Image processing;***

## 1. Introduction

Image denoising is a fundamental visual computing problem with numerous practical applications. For instance, noise in Monte Carlo (MC) rendering is a byproduct of the inherent variance in the MC estimator, and the noise magnitude is inversely proportional to the invested computation time. Rendering can be accelerated by using a denoiser, which processes a noisy initial image produced by a renderer with the help of additional scene features, and removes any remaining noise in the image. In contrast, noise in photographs is an inevitable byproduct of the mechanics of camera sensors, and can have a noticeably adverse effect on image quality especially in low light shots. Raw sensor noise consisting of "shot" and "read" components can be approximated by simple statistical distributions with reasonable accuracy. However, such approximations no longer work as the imaging pipelines of consumer cameras grow more complex over time. As such, an important challenge for practical image denoising is robustness against various noise characteristics resulting from real-world sources.

As with most low-level vision problems recent work in denoising focused on using data-driven methods that involve using various types of deep networks. Deep denoising networks often are trained for a specific type of noise with a narrow range of parameters, such as additive white Gaussian noise with a certain magnitude. Such specialized denoisers were initially outperformed by older patch-based methods when tested on datasets captured with real-world cameras, mainly as results of the difference between noise parameters they were trained for and the noise parameters at testing time. Blind denoising is a common strategy to generalize performance over a certain range of noise parameters by training with a diverse dataset. The *fundamental tradeoff of blind denoising*, however, is that robustness comes at the cost of overall denoising quality.

In this work we propose a new approach for designing denoisers for non-synthetic images that is robust to changes in noise characteristics. Our method does not follow the classical blind denoiser paradigm of training a single network using a diverse training dataset, nor does it require explicit camera pipeline models or additional components for estimating noise parameters. Instead, we propose a two-stage approach, where an input image with arbitrary noise characteristics is first processed by a number of *specialized denoisers*, that are each trained for a specific narrow range of noise parameters. We then introduce a second network that uses the outcome of the specialized denoisers as features to produce a denoising kernel for the specific characteristics of the noise present in the input image. Our method differs from previous kernel-predicting denoising architectures such as [MBC*18], in that our method focuses on forming accurate spatial kernels using specialized denoisers as extra feature channels, instead of relying on temporal information provided by a burst sequence.

## 2. Related Work

Prior to the wide scale use of convolutional neural networks (CNN) in image denoising, patch-based methods such as [ZW11] have been known to produce high quality results in practice. Algorithms such as BM3D [DFKE07] have been popular ever since and are known to work well for removing non-parametric noise that could for instance be produced by a camera sensor. In fact, the emergence of datasets of camera-captured images with sensor noise and the corresponding clean images [ALB18] revealed that deep

networks trained with synthetic data were outperformed by older methods such as BM3D. DnCNN [ZZC*17] reported for the first time a significant improvement over classical patch-based methods, and is still being considered the state-of-the-art solution for removing additive Gaussian noise. Other techniques utilized burst image sequences to exploit temporal information [MBC*18], which can potentially enhance denoising quality but requires careful attention to avoid ghosting artifacts. In [PCL*20], a multi-level wavelet residual network architecture and a progressive training scheme has been proposed for denoising. In [YZZM20], a dual adversarial network architecture has been proposed for image denoising as well as noise pattern generation by learning the joint distribution of the clean-noisy image pairs.

Recently, methods based on CNNs have also been successfully applied for denoising of MC renderings. Most related to our method are kernel predicting convolutional network (KPCN) based denoisers that express denoised pixel colors as a weighted linear combination of their neighboring pixels [BVM*17, VRM*18]. One key difference to denoising of non-synthetic images is that additional scene features can directly be obtained from the renderer with very low computation overhead (such as albedo, depth, surface normals, etc.). These scene features typically capture information about important structures in the image at much lower noise-levels than in the color image and help denoisers to distinguish noise from signal [ZJL*15]. Instead of relying on specialized equipment such as an RGB-D camera for acquiring additional data channels, we instead propose a novel way of generating features that augment the performance of KPCNs.

## 3. Robust KPCN

Our method utilizes a user defined number of **specialized denoisers**, each trained exclusively using noisy images with specific set of noise parameters $\{\lambda_0, \lambda_1, \ldots, \lambda_s\}$. At test time a specialized denoiser typically performs well if the noise parameters of the test image are similar to those of the training data. However as the noise parameters start to diverge from training data, testing performance drops rapidly. We introduce a **KPCN generalizer** to alleviate this inability of specialized denoisers to generalize over noise parameters. The KPCN generalizer takes as input not only an input image with some arbitrary noise parameters $\lambda_a$, but also utilizes the outcome of a set of specialized denoisers given the noisy input image, which we refer to as **denoised-image features**. For computational efficiency reasons we often want to keep the number of denoised-image features low. As such, in practice we expect the noise parameters $\lambda_a$ of an input image *not* to match the noise parameters $\{\lambda_0, \lambda_1, \cdots \lambda_s\}$ that the specialized denoisers were trained for. Our method therefore relies on the KPCN generalizer to estimate a suitable denoising kernel for an input image with noise parameters $\lambda_a$ given the denoised-image features. This way we achieve consistent denoising quality over a wide range of noise parameter, and hence we refer to our final method as **Robust KPCN** denoiser.

### 3.1. KPCN Generalizer

We represent the noisy input color image as a vector, which we denote as $\mathbf{x} \in \mathbb{R}^3$. We treat the denoised-image features as additional

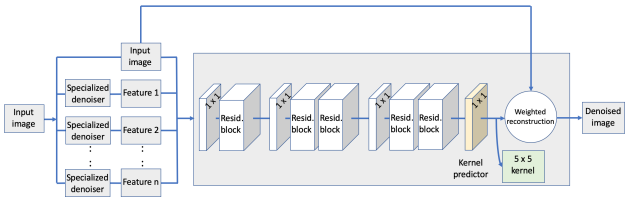


**Figure 1:** *Architecture of a KPCN. See text for discussion.*

channels $\mathbf{f}$, that consist of individual feature maps $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_s\}$. The KPCN generalizer takes as input the tuple $\{\mathbf{x}, \mathbf{f}\}$. Our objective is to find a model that minimizes the average distance between estimated denoised images $\hat{\mathbf{x}}$ and the corresponding noise-free ground-truth images $\mathbf{y}$. Formally, we express our model as $\hat{\mathbf{x}} = d(\{\mathbf{x}, \mathbf{f}\}; \hat{\theta})$, where $d$ denotes our denoiser with parameters $\hat{\theta}$. We compute the parameters $\hat{\theta}$ in a supervised setting using the dataset $\{\{\mathbf{x}^1, \mathbf{y}^1\}, \{\mathbf{x}^2, \mathbf{y}^2\}, \ldots, \{\mathbf{x}^n, \mathbf{y}^n\}\}$. Specifically, our objective is:

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} l(\mathbf{y}, d(\{\mathbf{x}, \mathbf{f}\}; \theta)), \tag{1}$$

where $l$ denotes the loss function and is discussed in Section /ref-sec:impl. The main difference between a KPCN and a direct prediction network is that the former estimates a $k \times k$ kernel of scalar weights around the neighborhood $\mathcal{N}(p)$ of a pixel location $p$. The weighted linear combination of the pixels in $\mathcal{N}(p)$ then gives the final pixel color prediction at location $p$.

Figure 1 illustrates the main building blocks of a KPCN architecture, where each residual block consists of two $3 \times 3$ convolutional layers bypassed by a skip connection. Thus, compared to the direct prediction of pixel colors, KPCNs introduce an intermediate kernel prediction step, which results in significant improvement in convergence speed as explored in detail in [VRM*18]. Moreover, prior experience has shown that KPCNs tend to generalize well to new data that has different characteristics than the training set [BVM*17]. In our method, the KPCN generalizer estimates a denoising kernel at pixel location $\mathbf{p}$ using the following identity:

$$\hat{\mathbf{x}}_p = d_p(\{\mathbf{x}, \mathbf{f}\}; \hat{\theta}) = \sum_{q \in \mathcal{N}(p)} w_{pq}\, \mathbf{x}_q, \tag{2}$$

where $w_{pq}$ denotes the normalized estimated weight at location $q$ belonging to the kernel at pixel location $p$. We normalize the kernel weights using a softmax function as in [VRM*18] so that they remain within $[0, 1]$.

### 3.2. Denoised-image Features

The idea of augmenting KPCNs with features that have low noise-levels has been previously explored for denoising of rendered images [ZJL*15, BVM*17, VRM*18]. These methods, in addition to mere colors, also exploit the additional types of scene information including albedo, surface normals, depth, visibility maps that are

often available in a rendering setting without significant extra effort. Figure 1 in supplemental material shows examples of typical feature channels that convey only partial scene information with respect to the corresponding color channel, but have the advantage of being often nearly noise-free and perfectly aligned with the color channel. Previous work showed that such features can be effectively utilized in a KPCN framework, which results in high quality denoising results even in challenging cases. A key difference when working with natural images is the absence of any additional scene information beyond the colors of an input image. Some extra information, such as depth, might still be captured using specialized equipment. But this approach brings in additional technical challenges such as alignment with the color image and dealing with missing or incorrect depth values, in addition to being less practical due to the extra hardware required. Moreover, it is unclear how to capture all the scene information that previous work in the rendering domain utilizes to achieve high quality denoising results. Another potential approach could be to extract scene information directly from color images. However, our experiments with recent work [ZSS*18] showed that the level of accuracy of the resulting predictions is not sufficient for our particular application. In fact, we found that using scene information predicted directly from input colors had a detrimental effect on denoising quality.

In our method we thus take a different approach and rely on features produced by a user defined set of specialized denoisers. Each specialized denoiser is typically trained for a specific noise type (e.g. noise produced by a specific camera sensor) or a narrow range of noise parameters (e.g. additive Gaussian noise with a specific magnitude). At testing time, an input image is processed by each of the specialized denoisers, producing the denoised-image features (Figure 2). The quality of each denoised-image feature, measured as the average distance from the corresponding clean image, varies depending on the noise characteristics of the input image. Thus, in a second step we introduce a KPCN generalizer that is trained to improve the final denoising quality by utilizing all the denoised-image features.

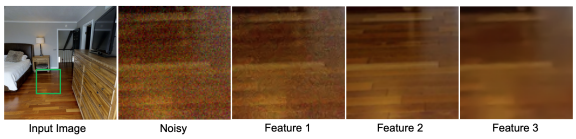


**Figure 2:** *Example denoised-image features that are generated by specialized denoisers given the noisy image as input.*

### 3.3. Model Implementation

While any denoiser can potentially be used to generate denoised-image features, in our experiments we used a U-Net architecture in our experiments that has originally been designed for image segmentation [RFB15]. In this work we directly adopt a modified version that has recently been shown to achieve comparable results to larger networks in image denoising applications [LMH*18] at faster training times.

We employed a two-step training procedure, where we first train a set of specialized denoisers, which we use to train a KPCN generalizer in a second step. We trained both the KPCN generalizers and specialized denoisers using patches of size $128 \times 128$, utilizing the Adam optimizer with an initial learning rate of $10^{-4}$, and later using dataset specific schedulers to decay the learning rate during the course of the training. All KPCN generalizers that were used to produce the results in this paper were trained with at mini-batch size 16 and used the mean absolute percentage error (MAPE) to assess the distance to a clean reference:

$$l(\mathbf{y}, \hat{\mathbf{x}}) = \frac{|\mathbf{y} - \hat{\mathbf{x}}|}{|\mathbf{y}| + \varepsilon}, \tag{3}$$

where $\varepsilon = 10^{-3}$ was introduced to avoid division by zero. The inference time of our whole method depends on the number and efficiency of individual specialized denoisers, as well as the resolution of the input image. Overall, Robust KPCN is runs efficiently at 320ms to process a $512 \times 512$ image single NVIDIA Titan X GPU.

| | PSNR / SSIM | | | | |
|---|---|---|---|---|---|
| Datasets | G4 | GP | IP | N6 | S6 |
| U-Net G4 | 32.99 / 0.876 | | | | |
| U-Net GP | | 31.03 / 0.726 | | | |
| U-Net IP | | | 32.78 / 0.730 | | |
| U-Net N6 | | | | 27.06 / 0.527 | |
| U-Net S6 | | | | | 22.91 / 0.432 |
| U-Net-B | 31.71 / 0.845 | 30.05 / 0.706 | 32.86 / 0.739 | 26.97 / 0.516 | 22.81 / 0.426 |
| Robust KPCN | **38.48 / 0.969** | **38.75 / 0.948** | **42.86 / 0.971** | **36.39 / 0.907** | **31.36 / 0.795** |

**Table 1:** *Average PSNR/SSIM results of various denoisers on SIDD benchmark dataset. See text for discussion.*

## 4. Camera Sensor Noise Experiments

To test our method we used the medium version of the recently published Smartphone Image Denoising Dataset (SIDD) [ALB18], which consists of 320 noisy and clean image pairs, obtained from 40 scenes under different lighting conditions using 5 smartphone cameras: Google Pixel (GP), Apple iPhone 7 (IP), Samsung Galaxy S6 Edge (S6), Motorola Nexus 6, (N6), LG G4 (G4).

We train a Robust KPCN denoiser by first using 150 image pairs from SIDD to train a separate specialized denoiser for each of the 5 cameras used to generate the images in the dataset. Each specialized denoiser is only trained with images that were captured using one of the 5 cameras. We adopted a U-Net model in order to implement the specialized denoisers, as it has been reported that this model is fast to train and shows good performance on denoising tasks [LMH*18]. The remaining 170 image pairs from SIDD are then used to train a KPCN generalizer which utilizes the specialized denoisers to generate 5 denoised-image feature channels.

Table 1 shows the average PSNR and SSIM values of Robust KPCN on the SIDD benchmark dataset consisting of 40 images. The columns of the table show the results from only images captured by specific cameras. The rows of the table show the results of various denoisers, where U-Net (G4, GP, IP, N6, S6) denote specialized denoisers trained for the corresponding cameras, U-Net-B denotes the blind denoiser trained with images captured with all cameras, and Robust KPCN denotes our full denoiser with 5 denoised-image feature channels that are obtained using the specialized U-Net denoisers (G4, GP, IP, N6, S6).

| Method | PSNR | SSIM |
|---|---|---|
| EPLL [ZW11] | 27.11 | 0.870 |
| KSVD-G [EA06] | 27.11 | 0.771 |
| KSVD-DCT [EA06] | 27.51 | 0.780 |
| CBDNet [GYZ*19] | 33.28 | 0.868 |
| DANet+ [YZZM20] | 39.43 | 0.956 |
| PT-MWRN [PCL*20] | **39.92** | **0.959** |
| Robust KPCN | 38.60 | 0.948 |

**Table 2:** *Average PSNR/SSIM results from SIDD sRGB Benchmark (Top published methods).*

We observe that the blind denoiser generalizes well over different types of camera sensor noise, but does so at the cost of decreased average PSNR and SSIM values with respect to corresponding specialized denoisers. Robust KPCN using 5 denoised-image feature channels improves upon both the specialized and blind denoisers by a large margin. We present a visual comparison in Figure 3.

Our method also compares favorably against previous denoising work in the SIDD benchmark challenge. In Table 2 we present our PSNR and SSIM results with other techniques as listed in [ALB18] that have been published at the time of submission. In terms of both PSNR and SSIM Robust KPCN is among the top methods in the benchmark.

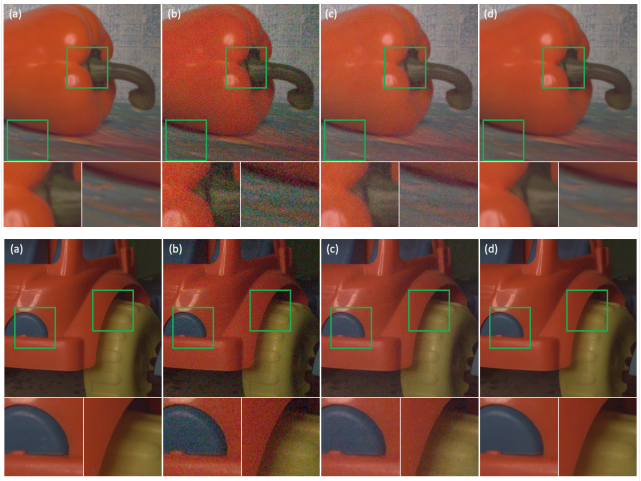


**Figure 3:** *Example denoising results from SIDD. (a) Ground truth, (b) Noisy image, (c) Specialized U-Net (d) KPCN with five U-Nets.*

## 5. Conclusion and Future Work

We presented a KPCN-based architecture for designing denoisers that are robust to noise parameters, and achieve near state-of-the-art denoising quality with real-world datasets. While we owe our inspiration of using a KPCN with feature channels to previous work in denoising Monte Carlo renderings [BVM*17], we think our new method of using denoised-image features within a KPCN framework could inspire more work back in the rendering community. We also think that extensions of our technique to video denoising could be an interesting direction for future research. From Table 1 we note that Robust KPCN performs significantly better than the specialized denoisers across the board. This suggests the specialized denoisers might be overfitting to the relatively small number of training samples available in the SIDD dataset. A thorough investigation of Robust KPCN on similarly limited datasets would be another interesting research direction.

## References

[ALB18] ABDELHAMED A., LIN S., BROWN M. S.: A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2018). 1, 3, 4

[BVM*17] BAKO S., VOGELS T., MCWILLIAMS B., MEYER M., NOVÁK J., HARVILL A., SEN P., DEROSE T., ROUSSELLE F.: Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Trans. Graph. 36*, 4 (July 2017), 97:1–97:14. 2, 4

[DFKE07] DABOV K., FOI A., KATKOVNIK V., EGIAZARIAN K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing 16*, 8 (Aug 2007), 2080–2095. 1

[EA06] ELAD M., AHARON M.: Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing 15*, 12 (Dec 2006), 3736–3745. doi:10.1109/TIP.2006.881969. 4

[GYZ*19] GUO S., YAN Z., ZHANG K., ZUO W., ZHANG L.: Toward convolutional blind denoising of real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019). 4

[LMH*18] LEHTINEN J., MUNKBERG J., HASSELGREN J., LAINE S., KARRAS T., AITTALA M., AILA T.: Noise2noise: Learning image restoration without clean data. In *2018 International Conference on Machine Learning (ICML)* (2018). 3

[MBC*18] MILDENHALL B., BARRON J. T., CHEN J., SHARLET D., NG R., CARROLL R.: Burst denoising with kernel prediction networks. *CVPR* (2018). 1, 2

[PCL*20] PENG Y., CAO Y., LIU S., YANG J., ZUO W.: Progressive training of multi-level wavelet residual networks for image denoising. *arXiv preprint* (2020). 2, 4

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. In *MICCAI* (2015), pp. 234–241. 3

[VRM*18] VOGELS T., ROUSSELLE F., MCWILLIAMS B., RÖTHLIN G., HARVILL A., ADLER D., MEYER M., NOVÁK J.: Denoising with kernel prediction and asymmetric loss functions. *ACM Trans. Graph. 37*, 4 (July 2018), 124:1–124:15. 2

[YZZM20] YUE Z., ZHAO Q., ZHANG L., MENG D.: Dual adversarial network: Toward real-world noise removal and noise generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*. August 2020. 2, 4

[ZJL*15] ZWICKER M., JAROSZ W., LEHTINEN J., MOON B., RAMAMOORTHI R., ROUSSELLE F., SEN P., SOLER C., YOON S.-E.: Recent advances in adaptive sampling and reconstruction for Monte Carlo rendering. *Computer Graphics Forum (Proceedings of Eurographics - State of the Art Reports) 34*, 2 (May 2015). 2

[ZSS*18] ZAMIR A. R., SAX A., SHEN W. B., GUIBAS L. J., MALIK J., SAVARESE S.: Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018). 3

[ZW11] ZORAN D., WEISS Y.: From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision* (Nov 2011), pp. 479–486. doi:10.1109/ICCV.2011.6126278. 1, 4

[ZZC*17] ZHANG K., ZUO W., CHEN Y., MENG D., ZHANG L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. In *IEEE Transactions on Image Processing* (2017). 2