TECHNISCHE
UNIVERSITÄT
DARMSTADT

MEASURING AND REMOVING REALISTIC IMAGE NOISE

A dissertation submitted to
TECHNISCHE UNIVERSITÄT DARMSTADT
Fachbereich Informatik

for the degree of
Doktor-Ingenieur (Dr.-Ing.)

presented by

TOBIAS PLÖTZ
M. Sc.

born in Wolfen, Germany

Examiner:	Prof. Stefan Roth, Ph. D.
Co-examiner:	Prof. Dr. Michael S. Brown, Ph. D.
Date of Submission:	13th of July 2020
Date of Defense:	1st of September 2020

Darmstadt, 2020

Plötz, Tobias : Measuring and Removing Realistic Image Noise
Darmstadt, Technische Universität Darmstadt,
Jahr der Veröffentlichung der Dissertation auf TUprints: 2021
Tag der mündlichen Prüfung: 01.09.2020

ABSTRACT

When capturing photographs with a digital camera, the resulting images are inherently affected by noise. Image denoising, *i. e.* the task of recovering the underlying clean image from a noisy observation, is fundamental to improve the perceptual quality, to help further visual reasoning, or to guide the optimization for more general image restoration tasks.

Since image noise is a stochastic phenomenon arising from different sources, such as the randomness introduced through the photon arrival process or the electric circuits on the camera chip, recovering the exact noiseless image is in general not possible. The challenge of the image denoising problem now arises by imposing suitable assumptions on both the formation process of the noisy image as wells as on the properties of clean images that we want to recover. These assumptions are either encoded explicitly within a mathematical framework that gives the denoised image as the solution of an optimization problem, or implicitly by choosing a discriminative model, *e. g.* a convolutional neural network (CNN), that is learned on training data comprised of pairs of clean and noisy images.

Having defined a denoising algorithm, it is natural to ask for assessing the quality of the output. Here, the research community by and large relies on synthetic test data for quantitative evaluation where supposedly noiseless images are corrupted by simulated noise. However, evaluating on simulated data can only be a proxy to assessing the accuracy on realistic images. The first contribution of this dissertation fills this gap by proposing a novel methodology for creating realistic test data for image denoising. Specifically, we propose to capture pairs of real noisy and almost noiseless reference images. We show how to extract accurate ground truth from the reference image by taking the underlying image formation process into account.

Since the image denoising problem is inherently ill-posed it is interesting to go beyond predicting a single possible outcome by additionally assessing the uncertainty of the prediction. Probabilistic approaches to image denoising naturally lend themselves for uncertainty prediction since they model the posterior distribution of denoised images given the noisy observation. However, inferring the quantities of interest, *e. g.* the marginal entropy at each pixel, is oftentimes not feasible. Our second contribution proposes a novel stochastic variational inference (SVI) algorithm that fits a variational approximation (Wainwright and Jordan, 2008) to estimate model-based uncertainty on the pixel level. We demonstrate that the resulting algorithm SVIGL is on par or even outperforms the strong baseline of SVI with the popular

Adam optimizer (Kingma and Ba, 2015) in terms of speed, robustness, and accuracy.

In this thesis we are also concerned with advancing the state of the art in terms of raw denoising accuracy. Currently, neural network based approaches yield the most powerful denoisers. Looking at more traditional methods, non-local approaches (Dabov et al., 2006) tend to be competitive. To combine the best of both worlds, in our third contribution we endow a strong CNN denoiser with a novel block matching layer, called neural nearest neighbors (N^3) block, for which we propose a fully differentiable relaxation of the k-nearest neighbor (KNN) selection rule. This allows the network to optimize the feature space on which block matching is conducted. Our N^3 block is applicable for general input domains as exemplified on the set reasoning task of correspondence classification.

While the aforementioned parts of this dissertation deal with the common case of a saturating camera sensor, *i. e.* intensity values increase up to a maximal value, we also consider a novel sensor concept called modulo sensor (Zhao et al., 2015) that is promising for high dynamic range (HDR) imaging. Here, pixel elements reset once they reach their maximal value. To obtain a plausible image we need to infer how often each pixel was reset during the exposure. In our fourth contribution we particularly want to reconstruct this information from multiple noisy modulo images. We propose to faithfully model the image formation process and use this generative model in an energy minimization framework to obtain a reconstructed and denoised HDR image, outperforming prior approaches to reconstruction from multiple modulo images.

ZUSAMMENFASSUNG

Bei der Aufnahme von Fotos mit einer Digitalkamera werden die resultierenden Bilder von Natur aus durch Rauschen beeinträchtigt. Bildentrauschung, also die Aufgabe, das zugrunde liegende saubere Bild aus einer verrauschten Beobachtung wiederherzustellen, ist von grundlegender Bedeutung, um die visuelle Qualität zu verbessern, weiteres visuelles Verstehen zu unterstützen oder die Optimierung für allgemeinere Bildwiederherstellungsaufgaben beeinzuflussen.

Da Bildrauschen ein stochastisches Phänomen ist, das von verschiedenen Quellen herrührt, wie zum Beispiel dem stochastischen Ankunftsverhalten von Photonen oder Rauschen in den elektrischen Schaltungen auf dem Kamerachip, ist es im Allgemeinen nicht möglich, das genaue rauschfreie Bild wiederherzustellen. Die Herausforderung des Bildentrauschungsproblems besteht nun darin, sowohl für den Entstehungsprozess des verrauschten Bildes als auch für die Eigenschaften der wiederherzustellenden rauschfreien Bilder geeignete Annahmen zu treffen. Diese Annahmen werden entweder explizit in einem mathematischen Modell codiert, in dem das entrauschte Bild als Lösung eines Optimierungsproblems gegeben ist, oder implizit durch Auswahl eines Unterscheidungsmodells wie zum Beispiel eines CNNs, das anhand von Trainingsdaten gelernt wird, die aus sauberen und verrauschten Bildpaaren bestehen.

Mit der der Entscheidung für einen Entrauschungsalgorithmus geht natürlich die Frage nach der Qualität seiner Ausgabe einher. Hier stützt sich die Forschung im Großen und Ganzen auf synthetische Testdaten zur quantitativen Auswertung, bei denen als rauschfrei angenommene Bilder mittels simulierten Rauschens verändert werden. Die Auswertung auf simulierten Daten kann jedoch nur eine Annäherung für die Genauigkeit auf realistischen Bildern liefern. Der erste Beitrag dieser Dissertation füllt diese Lücke, indem er eine neuartige Methodik zur Erstellung realistischer Testdaten für das Entrauschen von Bildern vorschlägt. Insbesondere schlagen wir vor, Paare von je einem echten verrauschten Bild und einem fast rauschfreien Referenzbild aufzunehmen. Wir zeigen, wie aus dem Referenzbild akkurate Ground Truth unter Berücksichtigung des zugrunde liegenden Bilderzeugungsprozesses extrahiert werden können.

Da das Problem der Bildentrauschung von Natur aus unterspezifiziert ist, ist es spanned, über die Vorhersage eines einzelnen möglichen Ergebnisses hinauszugehen, indem zusätzlich die Unsicherheit der Vorhersage bewertet wird. Probabilistische Ansätze zur Bildentrauschung eignen sich direkt für die Vorhersage von Unsicherheiten, da sie die a-posteri Verteilung der entrauschten Bilder gegeben der ver-

rauschten Beobachtung modellieren. Jedoch ist die Inferenz, *z. B.* der Marginalentropie an jedem Pixel, oft nicht möglich auf. Unser zweiter Beitrag schlägt einen neuartigen SVI-Algorithmus vor, der eine Variationsverteilung (Wainwright and Jordan, 2008) berechnet, um die modellbasierte Unsicherheit auf Pixelebene abzuschätzen. Wir zeigen, dass der resultierende SVIGL-Algorithmus in Bezug auf Geschwindigkeit, Robustheit und Genauigkeit die starke Vergleichsmethod von SVI kombiniert mit dem beliebten Adam-Optimierer (Kingma and Ba, 2015) erreicht oder sogar übertrifft.

In dieser Arbeit beschäftigen wir uns auch damit, den Stand der Technik in Bezug auf die Genauigkeit der entrauschten Bilder zu verbessern. Derzeit liefern Ansätze basierend auf neuronale Netzen die besten Ergebnisse und von traditionelleren Methoden können vor allem nicht-lokale Ansätze (Dabov et al., 2006) damit mithalten. Um das Beste aus beiden Welten zu kombinieren, kombinieren wir in unserem dritten Beitrag einen starken CNN-Entrauscher mit einem neuartigen Block-Matching-Layer, dem so genannten N^3-Block (neuronale nächste Nachbarn), für den wir eine vollständig differenzierbare Relaxation der KNN-Auswahlregel präsentieren. Dies ermöglicht es dem Netzwerk, den Merkmalsraum des Block-Matchings zu optimieren. Unser N^3-Block ist für allgemeine Eingabedomänen anwendbar. Das zeigen wir am Beispiel der Klassifizierung von Bildpunktkorrespondenzen, wobei das Netzwerk auf mengenwertigen Eingaben operiert.

Während sich die vorgenannten Teile dieser Dissertation mit dem Bildern eines saturierenden Kamerasensors befassen, *d. h.* die Intensitätswerte steigen nur bis zu einem Maximalwert, betrachten wir auch das neuartige Sensorkonzept eines Modulo-Sensors (Zhao et al., 2015), das für die HDR-Bildgebung vielversprechend ist. Hier werden Pixelwerte zurückgesetzt, sobald sie ihren Maximalwert erreicht haben. Um ein plausibles Bild zu erhalten, muss rekonstruiert werden, wie oft jeder Pixel während der Belichtungszeit zurückgesetzt wurde. In unserem vierten Beitrag rekonstruieren wir diese Informationen aus mehreren verrauschten Modulobildern. Unsere Methode basiert auf einem generativen Modell des Bilderzeugungsprozesses. Das rekonstruierte und entrauschte HDR-Bild erhalten wir anschließend durch Energieminimierung und wir zeigen, dass wir so die Genauigkeit gegenüber existierenden Ansätzen zur Rekonstruktion aus mehreren Modulo-Bildern verbessern.

ACKNOWLEDGMENTS

I am not a person of many words but nevertheless I want to take this opportunity to express my gratitude towards some of the people that I met in the recent years and that supported and influenced me during my PhD. First and foremost I want to thank Stefan Roth for sparking my interest in computer vision topics, for supervising me during my research and for all the discussions on technical topics where he challenged me to express my ideas in a sound and understandable way. I want to thank my many fellow PhD students and colleagues Thorsten, Kevin, Stephan, Junwha, Nikita, Nicole, Horst, Nils and especially my office mates Anne, Uwe, Jochen, and Faraz. These were the people who brought color to the everyday lab life and who were always ready to give a helping hand. I want to thank my family and friends just for all the good times and their non-technical view on things. And I want to thank my wife Daniela for her support and for giving me the chance to remind myself everyday that warm and trustful human relationships are much more important to me than research, papers and benchmark numbers.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AD analogue-to-digital

ADMM alternating direction method of multipliers

AEPE average end point error

AWGN additive white Gaussian noise

CCD charge coupled device

CMOS complementary metal-oxide-semiconductor

CNN convolutional neural network

CRF conditional random field

DNN deep neural network

EM expectation maximization

EXIF exchangeable image file format

FoE fields of experts

GAN generative adversarial network

GCRF Gaussian conditional random field

GL gradient linearization

GPU graphics processing unit

GMM Gaussian mixture model

GSM Gaussian scale mixture

HDR high dynamic range

HQS half-quadratic splitting

KL Kullback-Leibler

KNN k-nearest neighbor

LDR low dynamic range

MAP maximum a-posteriori

MLP multi-layer perceptron

MRF Markov random field

MSE	mean squared error
NLM	non-local means
PCA	principal components analysis
PSNR	peak signal-to-noise ratio
RBF	radial basis functions
ReLU	rectified linear unit
RMSE	root mean squared error
SGD	stochastic gradient descent
SISR	single image super-resolution
SSIM	structural similarity
SURE	Stein's unbiased risk estimator
SVI	stochastic variational inference
SVM	support vector machine
TOF	time-of-flight
TNRD	trainable non-linear reaction diffusion
TV	total variation
VAE	variational auto-encoder
VI	variational inference
VST	variance stabilizing transformation

1 INTRODUCTION

CONTENTS

Pictures are an invaluable vehicle to express thoughts, to document important events or to display a distorted version of reality in artistic ways. Since the advent of analogue cameras that were pioneered in 1816 by Joseph Nicéphore Niépce people were able to replace the daunting task of manually painting a scene by leveraging incident light hitting a film or sensor to form an accurate and realistic memory of the scene. The first cameras used analogue film, *i. e.* a thin layer of light-sensitive chemicals, to record light and required the post-hoc process of "developing" the raw negative into a positive. Technological advances in the 1970's allowed to directly measure the amount of incoming light on an array of electronic sensing units – called pixels – and store the resulting image in a digital format such that it can later be processed by a computer. In the sequel digital cameras revolutionized the field of photography by decreasing the effort and cost required to take a single image. Nowadays, cameras are a commodity and can be found in millions of smartphones, in cars, at industrial sites and many other places.

The abundance of visual data calls for an automated interpretation and analysis. Computer vision methods try to meet this need, *e. g.* by detecting or segmenting objects and categories in a scene (He et al., 2017), tracking people over longer time frames (Benfold and Reid, 2011), or establishing correspondences between key points in different images (Lowe, 2004). However, images might be affected by degradations and imperfections such as image noise, blur, or defunct

(a) (b) (c)

Figure 1.1. Three images spanning the history of photography. (a) shows one of the first fixed photographs taken on a metal plate with Bitumen being the light sensitive material. (b) is an example of classical analogue photography where film is illuminated during exposure. (c) shows an image taken with a modern digital smartphone camera.

pixels thus impairing both the application performance of the aforementioned tasks as well as the perceptual quality of the images. Some degradations can be countered by better camera hardware, *e. g.* to obtain a higher resolution of the images, while some other degradations, *e. g.* blur due to camera shake, can be avoided at capture time by carefully handling the acquisition device. Image noise, on the other hand, is inherent in any imaging process due to the stochastic nature of photons. Furthermore, as can be seen in Fig. 1.1 the characteristics of image noise depends on the imaging modality. While "View from the Window at Le Gras" shown in Fig. 1.1a – one of the earliest photographs – shows strong salt-and-pepper artifacts, images taken on analogue film are usually affected by film grain noise Fig. 1.1b. In digital cameras a major source of noise comes from the processing involved in the electron-to-digit conversion pipeline. However, even with ever better electronic circuits that reduce noise within the camera sensor, we still can not avoid noise due to the stochastic arrival process of photons hitting the image sensor. Hence, studying image noise and image denoising, *i. e.* the restoration of noisy images, is and will remain an important topic in computer vision research.

1.1 CHALLENGES

Image noise is an inherently stochastic quantity. Hence, in general we can not hope to *separate signal from noise* exactly from observing just a single image since many clean images might have given rise to the observed noisy image. This dilemma is usually approached by adding prior assumptions, both about the statistical distribution of clean images as well as about the nature of the noise. Prior knowledge about the noise distribution can either substantiate as an explicit model of the noise distribution, *e. g.* see Chapter 4, or as a simulation process. The later can be employed to produce pairs of clean and synthetic noisy images that can subsequently serve to train a discriminative model, *e. g.* see Chapter 5.

MEASURING IMAGE NOISE. In order to empirically study the distribution of image noise we need to collect multiple noisy measurements. For example, this can be done by either aggregating noisy samples spatially or temporally, which requires either picturing scenes with locally constant intensity or by capturing multiple images of the same scene. However, even when having access to plentiful noisy observations we might not be able to recover the underlying clean signal, *e. g.* by averaging the observations. For example, the noise distribution of a regular camera with saturating pixels is not zero-mean (Foi, 2009). Hence, we need to define prior assumptions about the noise distribution, usually by deriving a statistical model of the image formation process. In Chapter 3 we carefully model the generative process of capturing a pair of images with a conventional saturating camera sensor, effectively allowing us to obtain accurate ground-truth for image denoising.

DISCRIMINATIVE DENOISING. Discriminative approaches based on CNNs have shown tremendous performance improvements for many computer vision applications over the past years, (*e. g.* He et al., 2017; Krizhevsky et al., 2012; Zhang et al., 2017a). One enabling factor in this development is the availability of large scale training data sets. These are necessary since CNNs are highly flexible machine learning models and hence are prone to overfitting. While there have been large scale datasets for tasks like classification (Russakovsky et al., 2015), detection (Everingham et al., 2012) or image segmentation (Cordts et al., 2015), obtaining pairs of noisy images and an accurate estimate of its noise-free counterpart is an active area of research. The acquisition pipeline presented in Chapter 3 provides means to obtain large scale training data sets for image denoising.

Moreover, recent neural network models for image denoising (*e. g.* Zhang et al., 2017a; 2018), which are based on local operations like convolutions and element-wise non-linearities, have saturated in their performance and further gains in reconstruction quality often come at the expense of greatly increasing the number of feature maps per layer, thus leading to an exploding number of parameters. In Chapter 5 we show how we can augment such local processing networks with non-local aggregation operations, thus allowing for a significantly enlarged as well as adaptive receptive field while increasing the parameter count only modestly. The resulting denoising network shows strong performance gains especially for images with repetitive structures.

PROBABILISTIC INFERENCE. Methods based on probabilistic inference provide an attractive alternative to discriminatively trained models when training data is scarce. Commonly, probabilistic inference involves a two-stage process. First, a model of the posterior distribution of the to-be-estimated quantity is derived – often by in-

voking Bayes' rule and subsequently modeling the generative forward process. Second, according to some decision rule, *e. g.* the Bayes estimator, inference procedures are applied to the posterior to derive an estimate of the unknowns. Due to the complex posterior distributions used in many computer vision tasks the latter step is often computationally intractable and thus motivates approximate inference on the real posterior. In Chapter 4 we describe a novel approach that drastically simplifies and accelerates the process of finding a good approximate posterior for common problems in low-level computer vision.

APPLICATION TO DIFFERENT SENSORS. While camera sensors based on saturating pixels are prevalent other sensor concepts exist as well. The modulo sensor pioneered by Zhao et al. (2015) is especially suited for HDR photography. Here, the captured image requires a reconstruction to recover the HDR image. However, the original reconstruction algorithm of Zhao et al. assumes a noise-free sensor, which can not exist in practice. In a previous work (Lang et al., 2017) we studied how noise affects the reconstruction and presented an algorithm for robustly reconstructing an HDR image from several noisy modulo images, thus being applicable in real world scenarios. However, the latter work does not attempt to denoise the reconstruction and hence leads to noisy results. To alleviate this, in Chapter 6 we present a joint reconstruction and denoising algorithm that further increases the accuracy of the obtained HDR reconstruction.

1.2 CAMERA SENSORS

In this thesis we understand a camera as a device that allows to capture a two dimensional projection of the surrounding scene by recording incident light on a sensor medium. Prior to digital cameras, different chemical substances were used, *e. g.* special kinds of asphalt that hardens when exposed to light, or silver halides that turn to metallic silver once they receive a sufficient exposure to light.

In this thesis we will focus on more modern digital cameras, where the sensing elements are laid out in a discrete two-dimensional array of individual pixel sites. Digital cameras rely on the photoelectric effect to produce free electrons in a metal layer from the energy carried by incoming photons. These electrons are accumulated in a capacitor during the exposure time of the image. Finally, the voltage of the capacitor is transformed into a digital signal with the help of an analogue-to-digital (AD) converter. The last step, called *readout*, can be implemented in different ways. charge coupled device (CCD) sensors transport the accumulated charge from every pixel site to the border of the sensor where the amplifier and AD converter reside, while complementary metal-oxide-semiconductor (CMOS) sensors do the

conversion at every pixel site individually and thus enable parallel and hence faster readout. In order to measure multi-chromatic intensities filters are placed in front of the pixel elements, thus making them specific to certain sub bands of wavelengths. Usually, these filters are arranged in a regular pattern of red, green, and blue filters that gets repeated across the entire sensor to form the color filter array. While we focus on monochrome or RGB images in this thesis, color filters can also be used to obtain more general multi-spectral images, or to reduce unwanted ambient light in time-of-flight (TOF) cameras.

The digital intensity values obtained with the above process are called *linear raw* values. A subsequent *camera processing pipeline* transforms these into a visually pleasing image that can be displayed or printed. This involves several steps, many of which are proprietary. Some core stages, however, exist in most camera processing pipeline. First, *white balancing* is employed in order to scale the intensities of red, green and blue pixels such that neutral colors are recovered correctly. In a second step the missing colors at every pixel get interpolated by a process called *demosaicing*. Thus every pixel gets assigned a full color value in the camera internal color space. Next, a *color space transform* converts them to a more standardized color space, *e. g.* the sRGB color space. Finally, some form of non-linear transformation is employed to transform the linear intensity values into perceptually more plausible ones. For an an extensive overview over the camera processing pipeline, we refer to (Karaimer and Brown, 2016).

Besides the saturating and modulo camera, which we study in this thesis, there are other camera models as well, like event cameras (Lichtsteiner et al., 2008) that measure changes of light intensity rather than integrating the incoming light. While our contribution for HDR image reconstruction from modulo images (Chapter 6) is specific to this kind of imaging modality, our contributions regarding image denoising algorithms can be transferred to other camera sensors when either a generative model of the image noise distribution is known (Chapter 4) or paired training data is available (Chapter 5).

1.3 MODELS OF IMAGE NOISE

All steps involved in measuring the amount of incident light either introduce noise by themselves or affect the statistical distribution of the measurement noise, mostly yielding a more complex noise distribution. In Chapter 2 we will explain that noise in a linear digital raw image of a conventional digital camera is distributed according to a clipped Poisson-Gaussian distribution. We will use this model in Chapters 3, 4 and 6 for either gathering ground truth images or to remove image noise. Note, that the clipped Poisson-Gaussian model for linear raw intensities is still comparably simple. Further stages of the processing pipeline will either introduce dependencies of the noise between

color channels (*e. g.* through color space transforms), introduce spatial dependencies between pixel sites (*e. g.* through demosaicing) or make the noise distribution skewed and non-Gaussian (*e. g.* through gamma correction). In essence, the resulting noise distribution quickly becomes very complex.

Despite the fact that the Poisson-Gaussian model accurately describes image noise in linear raw images, much of the denoising literature is rather concerned with removing additive white Gaussian noise (AWGN). The AWGN model oftentimes leads to a simpler mathematical treatment of the denoising problem and it can also partly be justified by the observation that a heteroscedastic Gaussian distribution can approximately be transformed into a homoscedastic Gaussian distributions by means of a variance stabilizing transformation (VST) (Foi, 2009). After the VST regular Gaussian denoising methods can be applied and an inverse of the VST will approximately transform the resulting image back to the original space. Following the literature, we will also consider AWGN for our novel neural network based approach to image denoising, *cf.* Chapter 5.

Other forms of noise, that we do not explicitly consider here, encompass salt and pepper noise which might be caused by hot or defective pixels, or fixed pattern noise which arises due to non-uniform characteristics of the pixel sites but which can be accounted for by calibrating the sensor.

1.4 IMAGE DENOISING

Let us turn to the task of actually restoring a clean image from a noisy observation. We will discuss three important directions of research in this area in order to highlight some key challenges. Please note that we do not aim at providing a full and exhaustive taxonomy of denoising techniques but rather want to give context to the contributions of this thesis.

1.4.1 *Blind vs. Non-blind Denoising*

The strength of image noise can vary drastically depending on the employed camera sensor and capture parameters. Most consumer grade cameras allow to explicitly set an analogue gain factor – often called *ISO value* in reminiscence to the light sensitivity of analogue film – that influences the variance of the noise distribution in a linear way. Clearly, images affected with strong noise need to be treated differently from images which are only mildly affected by noise. Hence, many image denoising algorithms have an explicit parameter representing the noise strength. We call these algorithms *non-blind* since they expect the true noise strength to be approximately known. In contrast *blind* denoising algorithms just receive the noisy image as

input. Internally they need to figure out the noise strength in order to denoise the image appropriately. Please note that, at least for the case of removing AWGN or Poisson-Gaussian noise, any non-blind algorithm can be complemented by a separate noise estimation stage in order to obtain a blind denoising method. Moreover, many modern cameras provide an estimate of the parameters governing the Poisson-Gaussian distribution of the linear raw values through exchangeable image file format (EXIF) tags. Hence, in this thesis we focus on the non-blind case for the denoising methods presented in Chapters 4 to 6.

For more complicated noise distributions like those that arise after the non-linear camera processing pipeline, it becomes harder to summarize the noise distribution with a small set of parameters. Hence, recent work tries to fit high-dimensional parametric models such as Gaussian mixture models (GMMs) to explain the noise distribution of a single image (Nam et al., 2016). To model even more complicated distributions of image noise, Chen et al. (2018b) train a generative adversarial network (GAN) on a set of images with related noise characteristics. Afterwards they train a standard non-blind denoising network on a dataset of synthetically corrupted images, where the noise is sampled from the generator. Our denoising approach described in Chapter 5 is suitable to serve as the non-blind denoising network and hence can be used in conjunction with the GAN modeling of noise in order to denoise images affected by arbitrary noise.

1.4.2 *Probabilistic Models*

Probabilistic approaches to image denoising follow a two-stage process. First, the observed noise image $\mathbf{x}$ is used to derive a posterior distribution $p(\mathbf{y} \mid \mathbf{x})$ over the clean image $\mathbf{y}$. Second, an estimation function condenses the posterior distribution to a single prediction of the clean image, thereby employing probabilistic inference or optimization techniques. Regarding the first step, there are two different approaches to arrive at a posterior distribution.

GENERATIVE APPROACHES define a probabilistic model that describes the process of generating the observed noisy image from the unknown clean image. The generative model gives rise to the posterior distribution by virtue of Bayes' rule:

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y})}{p(\mathbf{x})} \propto p(\mathbf{x} \mid \mathbf{y})p(\mathbf{y}). \tag{1.1}$$

Here, $p(\mathbf{y})$ denotes the *prior* distribution over clean images while $p(\mathbf{x} \mid \mathbf{y})$ denotes the *likelihood* of observing the noisy image $\mathbf{x}$ when $\mathbf{y}$ is the clean image. A generative model has the advantage that we can synthesize data as long as we can sample from both the likelihood

and prior. Moreover, the prior can be reused for different tasks, *e. g.* inpainting or super resolution, by just changing the observation likelihood accordingly. Similarly, a generative approach is attractive for solving multiple related tasks that just differ in the parameters of the likelihood. For example, we can obtain different denoising models by changing the noise strength parameter of the likelihood.

DISCRIMINATIVE APPROACHES The versatility of generative models is a double-edged sword. Defining a good prior distribution that accurately captures the distribution of clean images is by itself a very hard endeavor. Traditional image priors remain simplistic by modeling the distribution of local filter responses (Roth and Black, 2011). However, sampling from these models reveals mostly cloudy structures without any semantic coherence in the image (Gao and Roth, 2012). But if we are not able to define an accurate prior distribution why should we bother with it in the first place? Discriminative probabilistic models, such as conditional random fields (CRFs) sidestep the need of a prior distribution by directly modeling the posterior $p(\mathbf{y} \mid \mathbf{x})$. To estimate the parameters of this distribution, a training set of pairs $(\mathbf{y}_i, \mathbf{x}_i)$ is necessary. In Chapter 3 we propose an acquisition pipeline for obtaining this kind of paired data.

INFERENCE. In order to arrive at an actual prediction, a probabilistic model needs to be accompanied by an estimator and an inference procedure for actually evaluating the estimator. Many estimators can be understood from the principle of *Bayes optimal estimation*. Here, we seek the prediction $\hat{\mathbf{y}}$ that minimizes the *expected loss* Δ assuming that the modeled posterior distribution accurately captures the distribution of the clean image given the observations

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}'} \mathbb{E}_{\mathbf{y} \sim p(\mathbf{y} \mid \mathbf{x})} \Delta(\mathbf{y}, \mathbf{y}') \tag{1.2}$$

$$= \arg\min_{\mathbf{y}'} \int \Delta(\mathbf{y}, \mathbf{y}') p(\mathbf{y} \mid \mathbf{x}) d\mathbf{y} \tag{1.3}$$

In general, the integral in Eq. (1.3) can not be computed explicitly. Instead one can obtain a Monte Carlo approximation to Eq. (1.3) by drawing samples from $p(\mathbf{y} \mid \mathbf{x})$ and minimize the resulting stochastic objective to solve for $\hat{\mathbf{y}}$. However, certain losses allow for an analytic expression of the integral. For example, when Δ corresponds to the $0-1$ loss $\Delta(\mathbf{y}, \mathbf{y}') = 1 - \delta(\mathbf{y}, \mathbf{y}')$, where δ denotes the Dirac delta function, Eq. (1.3) can be formulated as

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}'} p(\mathbf{y}' \mid \mathbf{x}) \tag{1.4}$$

This so-called maximum a-posteriori (MAP) estimator is a popular choice due its conceptional simplicity (Roth and Black, 2009). The

inference procedure is then an optimization of the posterior probability, usually carried out as a minimization of the negative logarithm of the posterior. In Chapter 6 we cast the problem of reconstructing a clean HDR image from multiple noisy modulo images in the MAP estimation framework.

When the integral in Eq. (1.3) can not be calculated in closed form and sampling the posterior is not a viable option neither, we can make a prediction based on a surrogate objective, where the true posterior $p(\mathbf{y} \mid \mathbf{x})$ is replaced with an approximating distribution $q(\mathbf{y})$. In order to reduce the approximation error we should choose $q(\mathbf{y})$ such that it is close to the true posterior $p(\mathbf{y} \mid \mathbf{x})$ *w. r. t.* some notion of distances between probability distributions. The problem of optimizing for the best $q(\mathbf{y})$ is called variational inference (VI), and it traditionally involves coordinate updates that are tedious to derive. To make VI more practical, we propose in Chapter 4 a stochastic optimization algorithm that only requires a linearization of the gradient of the log posterior *w. r. t.* the clean image $\mathbf{y}$, *i. e.* the gradient can be written in the form

$$\nabla \log p(\mathbf{y} \mid \mathbf{x}) = \mathbf{A}_{\mathbf{y}}\mathbf{y} + \mathbf{b}_{\mathbf{y}}, \tag{1.5}$$

where $\mathbf{A}_{\mathbf{y}}$ and $\mathbf{b}_{\mathbf{y}}$ are a matrix and a vector, respectively, that can depend on $\mathbf{y}$. We show that the gradient of typical log posterior distributions in low level computer vision problems can be linearized in a straightforward way and that the resulting stochastic optimization of $q(\mathbf{y})$ is faster than competing stochastic optimization techniques.

1.4.3 *Empirical risk minimization*

In recent years the field of computer vision has been revolutionized by learning based approaches hinging on the principal of empirical risk minimization. Ideally, we want to find an optimal mapping $f \in F$ from a family of functions F such that the expected loss on the data distribution $p_{\text{data}}(\mathbf{y}, \mathbf{x})$ gets minimized:

$$\hat{f} = \arg\min_{f \in F} \mathbb{E}_{p_{\text{data}}(\mathbf{y},\mathbf{x})} \Delta(\mathbf{y}, f(\mathbf{x})). \tag{1.6}$$

In practice, we can not optimize Eq. (1.6) directly as we do not have access to the data distribution p_{data} in an analytic way. Instead p_{data} gets characterized by a set of samples, *i. e.* pairs of clean and noisy images $(\mathbf{y}_i, \mathbf{x}_i) \sim p_{\text{data}}, i = 1, \ldots, N$. Moreover, the family of functions F is usually parameterized with a set of parameters θ. Thus, Eq. (1.6) can be reformulated as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \Delta(\mathbf{y}_i, f_\theta(\mathbf{x}_i)) + \mathcal{R}(\theta) \tag{1.7}$$

where the first term denotes the *empirical risk* on the training set and $\mathcal{R}(\theta)$ is a regularization term that penalizes overly complex mappings f. Regularization is necessary to avoid overfitting on the finite set of training samples. Empirical risk minimization is an attractive paradigm since the functions f_θ are usually easy to compute instead of requiring a demanding inference procedure as with probabilistic approaches.

Up until seven years ago the prevailing choice for the family of parametric functions F_θ were linear models on hand-selected feature spaces, like the popular support vector machine (SVM) approach (Schölkopf et al., 2002). They keep the optimization problem in Eq. (1.7) convex and performed well even when the training set is small. However, with the availability of large datasets and the increasing compute capacity of modern graphics processing units (GPUs) deep neural network approaches became a viable option. Neural networks are usually heavily parameterized functions and result in a highly non-convex optimization of Eq. (1.7). Especially CNN models have led to significant accuracy gains across the whole field of computer vision (*e. g.* He et al., 2017; Krizhevsky et al., 2012; Zhang et al., 2017a). Common CNN models for image restoration tasks (*e. g.* Mao et al., 2016; Zhang et al., 2017a) mostly consist of stacked convolutional layers and element-wise non-linearities. In Chapter 5 we show how these models can be augmented with non-local processing layers, thus improving upon state-of-the-art denoising performance.

1.4.4 *Local vs. Nonlocal Approaches*

Many denoising algorithms look only at small image regions around a pixel in order to denoise it. For example, simple Gaussian or median filtering chooses the denoised pixel based on a small neighborhood. Many generative models are local as well in that the prior is comprised of local filters and the likelihood is usually assumed to be pixel-wise independent. Although the prior nevertheless couples pixels across the whole image, the strength of the dependency as measured by the correlation of pixels quickly falls off with increasing distance. This is not necessarily bad, since natural images exhibit the same kind of diminishing statistical dependence the greater the distance between pixels gets. Denoising methods based on CNNs are also mostly local (*e. g.* Mao et al., 2016) as they are comprised of convolutional filtering and element-wise non-linearities. They can improve upon generative models with shallow image priors though, by deepening the network, thus expanding the receptive field of pixels in the output layer.

On the other hand, non-local methods try to infer a denoised pixel by aggregating observations from similar image regions. This follows the principle of *self-similarity* and in fact Zontak and Irani (2011) have shown that patches similar to a reference patch are more likely to

be found within the same image than within different images. This observation is exploited in *non-local* approaches to image denoising, such as the non-local means algorithm (Buades et al., 2005a) or the popular BM3D method (Dabov et al., 2006). In the latter, a KNN selection is applied to every patch in an image to find a set of similar patches. The distance function for measuring the similarity is usually an L_2 distance either in the pixel domain or in the domain of wavelet coefficients after thresholding. Having found a set of similar patches, collaborative filtering techniques are used to obtain a denoised version of each query patch.

It is now appealing to combine these two methodological approaches in a single denoising model. For CNN denoising this has been approached by discriminatively training the collaborative filtering part to maximize denoising accuracy (Lefkimmiatis, 2017; 2018; Yang and Sun, 2018). However, the search for similar patches is still kept a static part of the network model and relies on a hand-chosen distance function. In Chapter 5 we show how we can instead relax the non-differentiable KNN selection rule to obtain a differentiable nearest neighbor selection. This in turn allows to define a fully trainable non-local module for CNN based denoising that surpasses the accuracy of other non-local approaches.

1.4.5 *Measuring denoising accuracy.*

We will measure the quality of a denoising algorithm by comparing the denoised image $\hat{\mathbf{y}}$ to the ground truth clean image $\mathbf{y}$. Throughout this thesis we report two metrics that are commonly used in the scientific community. First, the PSNR which is given by

$$\text{PSNR}(\hat{\mathbf{y}}, \mathbf{y}) = 10 \log_{10}\left(\frac{\mathbf{y}_{\text{max}}^2}{\text{MSE}(\hat{\mathbf{y}}, \mathbf{y})}\right), \tag{1.8}$$

where $\mathbf{y}_{\text{max}}$ is the maximal intensity value that a clean image can attain and MSE denotes the mean squared error. Since the PSNR is closely related to the mean squared error, the optimal denoised image for this metric is close to the posterior mean of the clean image. Thus optimizing for PSNR often yields overly smooth images that are not perceptual plausible. To alleviate a part of this problem, we look at the the structural similarity (SSIM) index (Wang et al., 2004) as a second metric which combines a luminance, contrast and structural term. We use the widespread parameterization that weighs each term equally allowing the SSIM within a local neighborhood to be expressed as:

$$\text{SSIM}(\hat{\mathbf{y}}, \mathbf{y}) = \frac{\left(2\mu_{\hat{\mathbf{y}}}\mu_{\mathbf{y}} + c_1\right)\left(2\sigma_{\hat{\mathbf{y}},\mathbf{y}} + c_2\right)}{\left(\mu_{\hat{\mathbf{y}}}^2 + \mu_{\mathbf{y}}^2 + c_1\right)\left(\sigma_{\hat{\mathbf{y}}}^2 + \sigma_{\mathbf{y}}^2 + c_2\right)}, \tag{1.9}$$

where $\mu_{\hat{\mathbf{y}}}, \mu_{\mathbf{y}}$ measure local mean intensities, $\sigma_{\hat{\mathbf{y}}}, \sigma_{\mathbf{y}}$ measure the local standard deviation of intensities, $\sigma_{\hat{\mathbf{y}},\mathbf{y}}$ measures the cross covariance between the two images, and c_1, c_2 are small constants. The local estimates of the SSIM are subsequently averaged over the whole image. The SSIM index and its multi-scale extension (Wang et al., 2003) thus go beyond pixel-wise comparisons and instead correlate local image statistics between ground truth and prediction.

1.5 THESIS OVERVIEW

Chapter 2 gives an overview over relevant literature on image denoising and the underlying assumptions about natural images. Our main technical contributions are presented in Chapters 3 to 6. Specifically, in Chapter 3, which is based on (Plötz and Roth, 2017)[1], we propose a methodology for capturing realistic pairs of noisy and clean images which we use to benchmark existing image denoising algorithms. Concerning probabilistic approaches to image denoising, Chapter 4 introduces a novel stochastic optimization algorithm for variational inference and we evaluate it in the context of removing Poisson-Gaussian image noise and denoising 3D point clouds. This chapter is partly based on (Plötz et al., 2018) [2]. In order to advance the state-of-the-art accuracy in image denoising, we rely on learning-based approaches. Specifically, in Chapter 5 we propose N^3Nets that marry common local processing in CNNs with non-local aggregation of information, while retaining a fully end-to-end trainable pipeline. N^3Nets have already been published in (Plötz and Roth, 2018). In Chapter 6 we extend our prior work on reconstructing HDR images from noisy captures of an low dynamic range (LDR) modulo camera (Lang et al., 2017) by considering a joint HDR reconstruction and denoising from multiple noisy modulo images.

1.5.1 *Contributions*

AN IMAGE DENOISING BENCHMARK USING REAL IMAGES. Traditionally, image denoising algorithms have been evaluated on artificial noise images, that were synthesized by adding white Gaussian noise to presumably clean images. However, this it is not clear how results obtained with this evaluation approach will generalize to images

1 Chapters 3 and 5 cite in a verbatim way corresponding text from (Plötz and Roth, 2017) and (Plötz and Roth, 2018), respectively, since I am the main authors on both papers.

2 Note on contribution: Anne S. Wannenwetsch and myself contributed equally to (Plötz et al., 2018). The derivation of the SVIGL algorithm as well as theorems and proofs were jointly developed by both of us. My contribution to the experimental evaluation is mainly given by the experiments on Poisson-Gaussian denoising in Section 4.5.2 as well as 3D point cloud denoising in Section 4.5.3. Those two sections cite the respective text of (Plötz et al., 2018) verbatim.

corrupted by real image noise. To bridge this gap, in Chapter 3 we develop a novel methodology obtaining realistic benchmark data for image denoising. Specifically, we propose to capture a noisy image and an almost noise-free reference image of the same scene by adjusting the ISO value and exposure time appropriately. To ensure accuracy of the ground truth we develop a careful post-processing pipeline that corrects for spatial misalignments, linear intensity changes and low-frequency residuals in the reference image. To that end, we develop a novel Tobit regression model that is especially well suited for regressing on images taken with a regular saturating image sensor. We use this methodology to capture the *Darmstadt Noise Dataset (DND)*, a novel image denoising benchmark featuring images taken with four different consumer cameras covering a wide range of sensor sizes. An evaluation of different denoising methods on our benchmark reveals that accuracy on denoising synthetic AWGN does not necessarily correlate with accuracy on real images. Spurred by this finding subsequent research focused on denoising in more realistic scenarios and we discuss the progress made since the initial release of our dataset.

The next two contributions focus on improving image denoising capabilities in two distinct aspects.

STOCHASTIC VARIATIONAL INFERENCE WITH GRADIENT LINEARIZATION. Most literature in image denoising is concerned in computing a single denoised image that is as close to the ground truth as possible. Since denoising is an ill-posed problem the predicted image will in general divert from the ground truth. In our second contribution in Chapter 4 we look at the question how the certainty associated with such a prediction can be quantified. Probabilistic approaches lend themselves naturally to this problem, however intractable inference limits practicability. Hence, we resort to variational inference. Particularly, in Chapter 4 we propose a novel stochastic variational inference algorithm, called stochastic variational inference with gradient linearization (SVIGL), that combines recent advances in the optimization of stochastic functions with the technique of gradient linearization that is known to provide fast and accurate MAP inference on posterior functions used in many low-level vision problems. SVIGL is easy to implement, requiring only the linearization of the log-posterior gradient, while being fast and robust. In terms of accuracy SVIGL is on par or even outperforms strong state-of-the-art approaches to SVI as we show for the applications of Poisson-Gaussian denoising, 3D point cloud denoising, and optical flow estimation.

NEURAL NEAREST NEIGHBORS NETWORKS. Our next contribution aims at improving raw denoising accuracy by combining the flexibility of recent neural network approaches with more traditional, but still competitive non-local methods to image denoising, *cf.* Chap-

ter 3. Specifically we aim at integrating non-local processing into neural networks. Here, the main technical challenge lies in the non-differentiability of KNN matching, which is at the core of non-local methods, thus preventing gradient-based optimization of the matching feature space. In Chapter 5 we propose a continuous and deterministic relaxation of the KNN selection rule that is differentiable *w. r. t.* the distance metric employed for matching and allows to recover original KNN selection as a limit case. With our relaxation we define a novel non-local processing layer, called *neural nearest neighbors block* (N^3 block), that is based on the principle of self-similarity and generalizes wide spread attention layers. We demonstrate the effectiveness of the N^3 block by inserting it into strong baseline CNNs, yielding N^3 networks that achieve significant gains for image denoising and outperform competing non-local approaches that conduct KNN selection on fixed feature spaces. Moreover, both our continuous relaxation as well as the N^3 block are domain agnostic and can be used on other input modalities than images. We exemplify this by showing strong accuracy improvements on the set-reasoning task of correspondence classification by merely inserting a single N^3 block into a state-of-the-art baseline network.

For our last contribution, we switch gears and leave the realm of regular saturating camera sensors.

JOINT DENOISING AND HDR RECONSTRUCTION FROM MULTIPLE MODULO IMAGES. While natural scenes often exhibit a high range of intensity values, digital sensors are necessarily limited in the dynamic range they can capture, thus motivating HDR reconstruction from multiple, bracketed exposures. Here, modulo sensors are an interesting alternative to regular saturating sensors as they maintain detail in bright areas of a scene. Recent multi-exposure reconstruction algorithms for the modulo sensor have shown robustness to image noise. However, they treat each exposure individually and do not specifically try to remove image noise, leading to suboptimal visual results. In Chapter 6 we propose to jointly reconstruct and denoise a series of modulo images on order to obtain a high dynamic range image. Therefore, we cast the reconstruction problem in a probabilistic framework and solve for the MAP estimate of the resulting posterior distribution. We show that our approach leads to significantly better reconstructed images for realistic scenes, outperforming the reconstruction method of Lang et al. (2017) in settings with medium to strong noise while not deteriorating reconstruction accuracy in scenarios with little noise.

2

BACKGROUND AND RELATED WORK

CONTENTS

Since image denoising is a fundamental problem in image restoration, the body of literature on this topic is too big to be reviewed exhaustively in this thesis. We will therefore concentrate on presenting core principles underlying many image denoising methods. From the zoo of denoising algorithms we will review a selected subset and show how the aforementioned principles manifest therein. Let us start by characterizing the image denoising problem.

DENOISING AS UNDER-CONSTRAINED PROBLEM. Let $\mathbf{y} \in \mathbb{R}^{n \times m}$ denote a clean and noiseless image. Here, we treat $\mathbf{y}$ as an 2-dimensional array of gray-level intensity values. We assume that clean images do not fill the whole space of $\mathbb{R}^{n \times m}$ but rather lie on a manifold $\mathcal{Y} \subset \mathbb{R}^{n \times m}$ that describes the set of possible images that occur in a certain application context. In this thesis, we usually treat $\mathcal{Y}$ as the set of "natural" images, *i. e.* images that can be encountered in natural or man-made environments by looking at the world around us. We moreover assume, that the images $\mathbf{y} \in \mathcal{Y}$ are clean and not affected by any degradation process. Since every imaging process is subject to noise, we will never observe an instance of $\mathcal{Y}$ but rather another image $\mathbf{x}$ which is formed by corrupting the clean image $\mathbf{y}$ by some noise process η_β:

$$\mathbf{x} = \eta_\beta(\mathbf{y}). \tag{2.1}$$

We assume that the noise process can be parameterized by β, *e. g.* parameters related to a Poisson-Gaussian noise model. In the literature, it is often assumed that the noise process adds homoscedastic white Gaussian noise with standard deviation σ to the clean image $\mathbf{y}$, *i. e.*

$$\mathbf{x} = \mathbf{y} + \mathbf{n} \text{ with } \mathbf{n} \sim \mathcal{N}(0, \mathbf{I}\sigma^2). \tag{2.2}$$

The problem of denoising can now loosely be defined as recovering the noiseless image $\mathbf{y}$ from its noisy observation $\mathbf{x}$. The noise process

is a stochastic function, giving rise to a distribution $p(\mathbf{x} \mid \mathbf{y})$ of noisy observations given the underlying clean image $\mathbf{y}$. The difficulty of the denoising problem arises from the fact that usually multiple clean images

$$\mathbf{y} \in \mathcal{Y}_\mathbf{x} \text{ with } \mathcal{Y}_\mathbf{x} = \{\mathbf{y} \in \mathcal{Y} : p(\mathbf{x} \mid \mathbf{y}) > 0\} \tag{2.3}$$

can give rise to the same noisy image. Recovering the true underlying clean image exactly and with perfect confidence is thus hopeless. The best that we can hope for is to precisely characterize the preset $\mathcal{Y}_\mathbf{x}$ but commonly one tries to obtain a denoised image $\hat{\mathbf{y}}$ that is close to $\mathbf{y}$ under some notion of image distance.

In order to predict $\hat{\mathbf{y}}$, at least two ingredients are necessary. First, we must make some assumptions on the manifold $\mathcal{Y}$ of clean images in order to steer the prediction towards a clean image. Second, we must have some knowledge about the noise process η_β in order to make sure that the denoised image is still consistent with the noisy observation. Traditionally, both of these issues have been treated in an explicit model based manner. Recently, however, data driven approaches utilize implicit knowledge about $\mathcal{Y}$ and η_β by looking at pairs of example images $(\mathbf{y}_i, \mathbf{x}_i)$ from a large database.

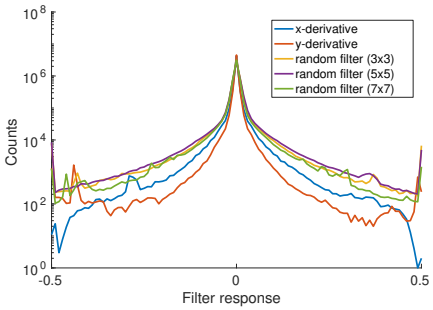
We will now first review different approaches to characterizing the properties of the natural image manifold $\mathcal{Y}$.

2.1 STUDIES OF NATURAL IMAGES

In the following we want to present three different principles that try to characterize the natural image manifold. First, we look at the statistical distribution of local filter responses that allow us to establish local dependencies between image pixels. Second, we look at scale invariant properties of images. Finally, we review work on the principle of self-similarity that describes the phenomenon that image structures tend to reoccur within an image.

LOCAL IMAGE STATISTICS. Studies of the marginal statistics of local filter responses have been among the first attempt of describing natural images. It has been found that the simple cells in the visual cortex, *i. e.* the first processing layer of our visual system, respond to an array of local stimuli that are separated spatially and in terms of frequency and orientation, thus resembling Gabor filters (Marĉelja, 1980). Driven by the idea that our visual system should be efficient in representing images that it encounters in every day life, Field (1987) proposed to analyze the marginal distributions of Gabor filters in natural images. He found that these distributions follow a power law in the frequency domain and that images can be represented sparsely, *i. e.* only a few Gabor filters are highly active at the same time. While Field showed that filters that are localized and band-limited lead to sparse activations, (Olshausen and Field, 1996) demonstrated in

(a)



(b)

Figure 2.1. The distribution of filter responses is non-Gaussian. Left: An example image of a bird. Right: Histogram of filter responses calculated on the example image for different filters. The distributions are sharply peaked at zero and exhibit heavy tails.

subsequent work that the reverse direction is also true. They wanted to study codes of natural images that leads to sparse activations. They did so by minimizing an energy comprised of a data term measuring how well an image can be represented by a linear combination of code-words, and a prior term favoring sparse activations. The learned filters indeed are oriented, band-limited and spatially localized. This is opposed to Fourier basis filters that can be derived from principal components analysis, where the goal is to maximize the activation variance, or energy. Moreover, (Olshausen and Millman, 2000) showed that sparsity of activations automatically emerges even when the prior over activations is a flexible mixture of Gaussians which could in principle also model non-sparse activations.

Sparsity also emerges as a property of the distribution of filter responses. Figure 2.1 shows an example image and histograms of the response distribution for several filters. We can observe that the distributions are non-Gaussian with a sharp peak at zero and heavy tails, meaning that filters are inactive most of the times while still having a significant probability of generating a strong response.

Going beyond modeling of marginal distributions of filter responses (Simoncelli, 1997; 1999) investigate the joint distribution of wavelet coefficients corresponding to wavelets with neighboring scale, orientation or spatial position. They show that coefficients from neighboring wavelets tend to be highly dependent and that a coefficient can be well predicted from a linear combination of its neighboring coefficients. Capturing this second order information is also helpful for synthesis. While matching only marginal distributions of wavelet coefficients between a target image and a synthetic image will lose the overall texture, texture patterns such as stripes or block structures can be reconstructed well when also preserving second-order correlations between wavelet coefficients.

To summarize, the principle of sparsely coding images is well motivated, both from a biological as well as an empirical point of view

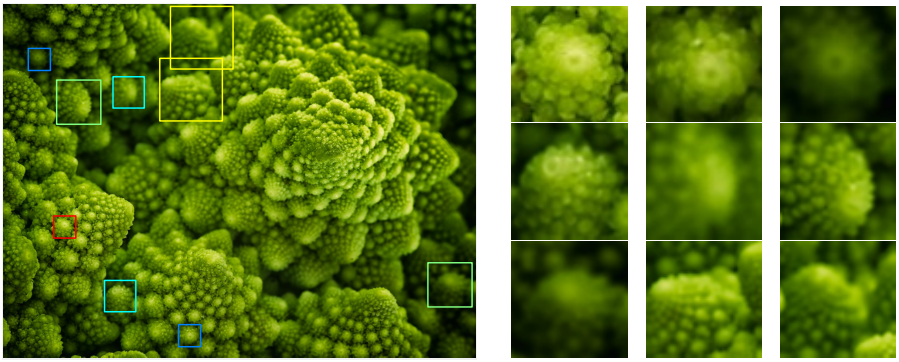
and has been used as prior assumption in many denoising algorithms, some of which will be discussed later in this chapter.

SCALE INVARIANCE. Another fundamental property of natural images is scale invariance of local statistics. Loosely speaking this means that statistics of the image should not change if we zoom in or out, and that image structures should appear at different scales. This assumption is well supported intuitively when looking at typical fractal like structures found in nature, for example the picture of the Romanesco plant in Fig. 2.1b. Also physical laws lead to emergence of scale invariant behavior (Coleman and Pietronero, 1992; Turcotte, 1995). Field (1987) showed that the amplitude of the power spectrum falls off approximately inversely proportional to the frequency which implies that energy is constant across scales. In a further work Field (1993) also introduced a notion of scale invariance of the phase spectrum where phases are found to be aligned across different frequencies. For example, an edge found at a certain scale in the image is likely to be also present in a lower or higher scale. This property is important to sparsely represent features like edges, blobs and other localized image structures. Ruderman (1994) further reviewed scale invariance properties of natural images.

Ruderman (1997) connected scale invariance of the power spectrum to scale invariance of the autocorrelation function in the spatial domain. They formulate a model of synthetic images which are comprised of independent, occluding object of constant intensity whose size follow a power law distribution. Image synthesized with this model exhibit the same scale invariance properties *w. r. t.* to the autocorrelation function as can be found in natural scenes.

More recently, Zoran and Weiss (2009) analyzed the kurtosis, *i. e.* the fourth standardized moment, of natural images and found that the kurtosis of filter responses obtained from clean images are approximately constant across scale. However, when corrupting images with independent noise, the kurtosis will drop for higher frequencies, effectively allowing to estimate the amount of noise that is affecting the image.

EXTERNAL VS. INTERNAL RECURRENCE. A third principle that characterizes the manifold of natural images is called self-similarity, referring to the phenomenon that image structures tend to reoccur within an image. Referring to Fig. 2.2 we can intuitively see that it is not very likely to encounter the intricate structure of the florets when looking at another random picture. However, within the photo of the Romanesco, the structure of the florets are abundant. The phenomenon of self-similarity was first quantitatively analyzed by (Zontak and Irani, 2011). They compared the probability of finding a match of a 5×5 patch within the same image to the probability of



(a) (b)

Figure 2.2. An example illustrating the principle of self-similarity. Left: An example image of a Romanesco fruit. The query patch is marked with a red rectangle. The other rectangles indicate the two closest matching patches per scale for four different scales where similarity is measured by normalized cross-correlation. Right: Close-ups of matching patches, with the query patch shown in the top left corner.

finding a similar patch within an external patch database created from different images. With this setup they found that the image-internal patch database is more expressive than a big external database in the sense that, for a certain query patch, the closest patch in the internal database is likely to be closer than the closest patch of the external database, with distance measured by the L_2 distance. This tendency is more pronounced when the mean gradient magnitude of the query patch is high, *i. e.* when the patch exhibits strong structures instead of uniform or smooth areas. They conclude that self-similarity is best cast as a non-parametric prior since any parametric scheme to summarize the image content is likely to discard the patches with high gradient magnitudes as they only occur rarely.

Let us now take a look at how these three principles manifest in actual denoising algorithm.

2.2 MODEL-DRIVEN APPROACHES

We first review denoising methods that formulate prior knowledge in terms of an explicit mathematical model.

SPARSE CODING We start by looking at approaches that connect to the findings of Olshausen and Field (1997) who demonstrated that sparse codes are a good representation for natural images. One prominent example in this category is the K-SVD algorithm (Aharon et al., 2006). Given a dataset of natural images, it learns to represent image patches as sparse combinations of an overcomplete dictionary, showing that the resulting dictionary is more efficient than hand-chosen bases. The K-SVD algorithm was subsequently integrated into a denoising algorithm for gray scale (Elad and Aharon, 2006) and color

denoising (Mairal et al., 2008). The basic K-SVD denoising algorithm was recently even more refined by Xu et al. (2018a) to accommodate different noise levels per color channel, approximately signal dependent noise and different strength of the regularization for different sparse coding coefficients. The resulting TWSC algorithm achieves very high accuracy for real color image denoising as demonstrated on our benchmark, *cf.* Chapter 3.

The previous techniques learn the coding dictionary from an external database of images. There are several approaches that also integrate image-internal information based on the principle of self-similarity into the sparse coding framework. In the LSSC algorithm proposed by Mairal et al. (2009) sparsity is not only promoted on a per-patch level but LSSC also favors similarity of coding coefficients within a group of similar patches. While LSSC implements group-level sparsity with a group lasso approach, the NCSR algorithm of Dong et al. (2013) is built on the assumption that coding coefficients within each group are clustered tightly around a prototypical set of coefficients. Recently, Xu et al. (2018b) have proposed to extend externally learned dictionaries with dictionaries derived from patch groups within the noisy image. They first represent external image patches as a mixture of Gaussians and compute the first leading eigenvectors of the covariance matrix as code words for each mixture component. Having assembled noisy patches into patch groups, each group gets assigned to the closest Gaussian and its corresponding dictionary is completed to an orthogonal basis under sparsity constraints.

MODELS OF LOCAL FILTER STATISTICS We now turn to denoising approaches that are based on modeling the statistics of local filter responses within an image. Wavelets are a popular choice for these filters and there is a large body of work that deals with regularizing their coefficients. The main idea is that large wavelet coefficients are likely to be caused by image signal whereas small coefficients are likely caused by noise. Hence the amplitude of the coefficient is reduced in order to remove the noise. The theoretical foundations of this wavelet shrinkage approach has been laid out by Donoho (1995) and subsequently put into practice in many works, *e. g.* (Chambolle et al., 1998; Simoncelli, 1999; Simoncelli and Adelson, 1996). It has also been noted that wavelet coefficients exhibit correlations across space and scale (Portilla and Simoncelli, 2000; Simoncelli, 1999). Particularly, the popular BLS-GSM approach of Portilla et al. (2003) models the local distribution of coefficients with Gaussian scale mixtures (GSMs) (Andrews and Mallows, 1974) and casts denoising in a Bayesian least squares framework.

While the previous approaches treat individual wavelet coefficients or groups thereof individually, Markov random field (MRF) based image priors (Geman and Geman, 1984) provide global regularization.

A very flexible and popular model is the fields of experts (FoE) model of Roth and Black (2011). There, the image prior is defined as a product of potential functions on overlapping local filter responses, thus coupling all pixels in a rigorous probabilistic framework. The parameters of the filters and potential functions can be learned from data. However, inference and learning in a FoE model is challenging as both require either approximate variational inference techniques or sampling. To speed up sampling (Schmidt et al., 2010) propose an efficient auxiliary variable Gibbs sampler for MRFs with potential functions parameterized as GSMs. For the case of removing AWGN this technique also allows to sample from the posterior distribution of clean images given a noisy observation, making the approach amenable to Bayes optimal denoising.

Learning of FoE models was further improved by Gao and Roth (2012). They show that the samples of the fitted FoE model accurately reproduce local image statistics found on natural images. Since MRF image priors are probabilistic models we can also assess the likelihood of a set of natural images in order to judge the quality of the prior. A main technical difficulty, however, is posed by the intractable partition function, *i. e.* the normalization constant. Bounds on the partition function of FoE models with GSM potentials are given by Weiss and Freeman (2007). As an alternative the partition function can be estimated from samples, *e. g.* by using annealed importance sampling (Neal, 2001).

Black et al. (1998) drew a close connection between MRF priors and anisotropic diffusion models (Perona and Malik, 1990). They found that the time discretized diffusion steps correspond to gradients of a log MRF prior with pairwise potential functions. They exploit this observation to robustify anisotropic diffusion by choosing robust potential functions, *e. g.* a Lorentzian. Moreover, there is a relationship between MRFs and variational methods that define the image on a continuous support (Schelten and Roth, 2011). A prior model similar to the FoE was proposed by Zhu and Mumford (1997). In contrast to Roth and Black (2011) they select filters from a filter bank and learn corresponding non-parametric potential functions in a minimax entropy framework.

NON-LOCAL APPROACHES A third major group of model-driven denoising algorithms is comprised of non-local methods based on the principle of self-similarity. One of the seminal works in this context was done by Buades *et al.* (Buades et al., 2004; Buades et al., 2005a; Buades et al., 2005b) where they proposed the non-local means (NLM) algorithm. Here, a noisy pixel is reconstructed as a weighted average of all other pixels, the weight being given as a function of the similarity of the respective local neighborhoods. As the calculation of similarities between all local neighborhoods becomes prohibitively

expensive with growing image size, in practice only pixels in a local window are considered in the weighted average. Thus, there is also a tight relationship between the NLM algorithm and the bilateral filter (Smith and Brady, 1997; Tomasi and Manduchi, 1998), the latter having a Gaussian range window and similarities are computed just by comparing pixel values. To avoid the restricting the size of the search window, Talebi and Milanfar (2014) propose to sparsely sample the full pixel-to-pixel weight matrix and then recover an approximation to the full matrix by virtue of the Nyström extension (Nyström, 1928).

Subsequent work on non-local denoising differs from the basic non-local means algorithm in two aspects. First, while the non-local means algorithm gives a non-zero weight to all patches within the search window of a noisy pixel, later work selects only a subset of patches that are most similar to the patch around the noisy pixel. This strategy is often referred to as *block matching* as for each noisy query patch a 3D volume of matched patches is computed. Second, while the non-local means algorithm uses a simple weighted average to non-locally aggregate information, subsequent work explores non-linear aggregation. For example, the very popular BM3D algorithm (Dabov et al., 2006) performs collaborative filtering on the 3D volume of matched patches by applying thresholding on wavelet coefficients. Afterwards the wavelet transform is inverted and the denoised image intensities get redistributed to the image domain. Other approaches try to find a basis that describe each patch group. The non-local Bayes algorithm of Lebrun et al. (2013) models each patch group as samples from a Gaussian distribution. For actual denoising the empirical means and covariances are used to obtain the mode of the Gaussian posterior of the clean image patch. Non-local Bayes relates to the PLOW algorithm (Chatterjee and Milanfar, 2012) that also models similar patches as coming from a Gaussian distribution but employs a more elaborate procedure for matching and aggregation.

Instead of projecting the noisy patches to linear subspaces learned from each patch group, one can also use an overcomplete basis onto which noisy patches are projected in a sparse coding framework. In (Mairal et al., 2009), sparse coding coefficients of all patches in a group are coupled with a group sparsity regularizer. Similarly, the NCSR algorithm (Dong et al., 2013) regularizes sparse coding coefficients of each patch group to be close to a set of mean coefficients. They solve the resulting optimization problem by iteratively updating the current sparse coding coefficients and subsequently the estimate of the mean coefficient. The WNNM algorithm (Gu et al., 2014) exploits the observation that, while sets of general image patches are not coded sparsely by a orthogonal and thus not overcomplete basis, patch groups of similar patches are more likely to live on a linear subspace. Hence, they fit a low dimensional basis to each patch group by minimizing a weighted form of the nuclear norm. Here, the individual singular

values are weighted according to their norm, as the authors claim that basis vectors with a large singular value are likely to carry clean image signals while basis vectors with a small singular value are likely to be caused by noise. Recently, WNNM has been extended to also handle color images (Xu et al., 2017a) where they especially show improved performance on denoising the real world sRGB images of our benchmark Chapter 3.

Other approaches try to combine offline learning of an image prior with online adaptation to specific images. For example, Xu et al. (2015) fit a GMM to an external database of image patches. For denoising a new image, patch groups are formed by block matching, matched to the closest mixture component and denoised in a sparse coding framework using the singular vectors of the covariance matrix as dictionary. This approach was further improved by Xu et al. (2018b). For forming the dictionaries they only take the leading singular vectors of the covariance matrix as basis vectors and complement them with orthogonal basis vectors learned from each patch group. They thus obtain image specific dictionaries that are subsequently used in a sparse coding setting to denoise patches within each patch group. Chen et al. (2015a) propose to cluster patches of a noisy image according to a previously learned GMM. They proceed by denoising each patch group in an (unweighted) nuclear norm minimization framework. These two steps are iterated until convergence, reaching accuracy comparable to the WNNM algorithm which uses standard block matching instead of clustering based on a pretrained GMM. Mosseri et al. (2013) complement non-local denoising with non-parametric external denoising. They show that denoising based on self-similarity is better than external denoising for noisy patches with little structure while denoising with the external patch database achieves better accuracy for patches which have strong variance of the image signal. They propose to combine both approaches by adaptively deciding whether to use internal or external denoising.

While a lot of research focuses on the *processing* of patch groups, the question of how the *matching* of patches is conducted has received surprisingly little attention. A study conducted by Deledalle et al. (2012) compares different similarity measures for image denoising with the non-local means algorithm under different noise distributions, *e. g.* Gamma or Poisson noise. They found that a similarity metric based on likelihood ratios works well in the considered cases. Moreover, an Euclidean similarity metric should be adapted to non-Gaussian noise by first applying a variance stabilizing transformation. Zontak et al. (2013) and Lotan and Irani (2016) show that patch matching is more reliable when not only considering the noisy patches by themselves but also their coarser-scale versions. The work of Frosio and Kautz (2019) looks at the statistic of patch distances of noisy patches that originate from the same clean patch. For Gaussian noise the expectation of

this distance is proportional to the noise standard deviation. They exploit this observation for patch matching by deeming those pairs of patches as similar that have an observed Euclidean distance close to its expectation. Patches that are too close in Euclidean space are thus not considered similar since the closeness is more likely due to similar noise patterns rather than similar image content. They use their statistical nearest neighbor selection within the non-local means framework for denoising and demonstrate a consistent gain over standard nearest neighbor selection. Our work on differentiable nearest neighbor selection in Chapter 5 complements the work on statistical nearest neighbors as it allows to additionally learn the feature space on which patch matching is conducted.

2.3 DATA-DRIVEN APPROACHES

In this section we review common strategies to design data-driven approaches for image denoising. We will start off by discussing work on learning flexible parametric models, *e. g.* a CNN or some other form of deep neural network (DNN), by training on a huge set of examples. Next, we will look at approaches that are rooted in classical inference procedures, such as MAP estimation in probabilistic models, whilst interpreting the process of inference itself as a prediction function that entails parameters which can be optimized. Lastly, we will look at methods that combine CNNs with ideas from non-local denoising methods.

PLAIN LOCAL DNN / CNN Zhang and Salari (2005) were among the first to use a convolutional neural network for image denoising. Specifically, they propose to train multiple three-layer networks that denoise wavelet coefficients corrupted by AWGN. Each network operates on one sub-band of the wavelet decomposition. Different to contemporary network architectures they employ a point symmetric activation function for the hidden layer. This function is linear around zero but then shrinks feature activations with high magnitude towards zero. Even though they train on just a single image, they achieve substantial improvements over competing, hand-crafted wavelet shrinkage methods.

In a more refined attempt to use CNNs for image denoising, Jain and Seung (2009) train a network with four hidden layers having 5×5 convolutional kernels and 24 feature channels each. The activation function is chosen as a sigmoid. They train on 24×24 patches and observe that bigger training patches did not lead to improved accuracy. They also applied layer-wise training, probably due to the now well-known *vanishing gradient problem* of deep sigmoid-activated networks. Nevertheless, they achieved a consistent gain over the generative FoE

model (Roth and Black, 2011) and BLS-GSM (Portilla et al., 2003), both quantitatively and regarding visual quality.

While these early approaches deliberately tried to keep the number of trainable parameters small in order to maintain computational practicality, Burger et al. (2012) wanted to analyze the accuracy of a patch based multi-layer perceptron (MLP) denoiser that is given a very high capacity and a large training database. Their model for denoising 17×17 patches has 4 hidden layers with 2048 features each, resulting in – even by today's standards – a huge network with more than 16 million parameters. After training for one month on a large training set of $150,000$ images, the model was able to perform on-par to BM3D (Dabov et al., 2006).

The result of Burger et al. (2012) was probably a bit discouraging given the immense computational effort required to train their MLP network. In the same year of 2012, CNNs started their still ongoing success in the computer vision community with Alex Krizhevsky *et al.* winning the ImageNet image classification challenge by a huge margin using a CNN based method (Krizhevsky et al., 2012). However, it took some more years until Zhang et al. (2017a) popularized CNNs also for image denoising. They proposed a network called DnCNN. The main differences to the early work of Jain and Seung (2009) can be summarized as follows. First, the DnCNN consists of 17 convolutional layers with 64 features each, thus being considerably deeper and wider. Second, in order to train such a deep network, Zhang et al. employed rectified linear unit (ReLU) activation functions (Glorot et al., 2011; Hahnloser, 1998) and batch normalization layers (Ioffe and Szegedy, 2015). Third, they use residual learning, *i. e.* instead of directly regressing the denoised output image, the DnCNN predicts the residual between the noisy and clean image. Combining these changes lead to a very simple yet effective network that outperformed competing approaches. They also showed that a single model is able to handle different noise levels if trained properly, a feature that is important for practical applications. In follow-up work Zhang et al. (2018) proposed FFDNet. Here, they quadrupled the size of receptive field by converting pixels within a 2×2 neighborhood into 4 input feature channels. Moreover, they have another input channel that corresponds to the input noise strength and can account for a spatially varying noise level. The recent WDnCNN (Bae et al., 2017) combines the basic architecture of DnCNN with the idea of denoising wavelet coefficients, similar to Zhang and Salari (2005). Moreover, having 320 feature channels the network is significantly wider than the baseline DnCNN. Their experiments show a consistent gain over DnCNN, but unfortunately the paper does not discern whether this is due to the wavelet denoising approach or due to the higher capacity of the network. Recently, Liu et al. (2017) proposed to apply the wavelet decomposition at multiple scales to further extend the receptive field.

Concurrently to DnCNN, Mao et al. (2016) proposed REDNet, a wider and deeper network. In contrast to DnCNN, REDNet does not use padded convolutions but instead has a contracting and expanding part. Additive skip connections between layers of the same resolution help to maintain high frequency details. In further work, Tai et al. (2017) introduced MemNet for image restoration tasks. It consists of multiple memory blocks, each of which being comprised of multiple residual blocks with weight sharing. The memory blocks are densely connected. They show slight performance improvements over REDNet. However, even though newer methods achieved accuracy gains over DnCNN, the latter remains the prevailing choice due to its good trade-off between a simply architecture and good accuracy.

UNROLLED OPTIMIZATION Another line of work is rooted in classical approaches that define the denoised image as the solution to an optimization problem:

$$\hat{\mathbf{y}} = \arg\min_{\mathbf{y}'} J(\mathbf{y}', \mathbf{x}) \tag{2.4}$$

In the context of MAP estimation the objective function J is given by an energy function, *i. e.* the negative log posterior, while more generally J is comprised of a data or reaction term favoring solutions close to the noisy image $\mathbf{x}$, and a prior or diffusion term that encodes assumption about the manifold of natural images. Problems akin to Eq. (2.4) are often non-convex. Hence, in practice they are solved with an iterative optimization or inference procedure that is run for T time steps:

$$\hat{\mathbf{y}} \equiv \hat{\mathbf{y}}^T \text{ with } \hat{\mathbf{y}}^{t+1} = f^t(\hat{\mathbf{y}}^t) \text{ and } \hat{\mathbf{y}}^0 \equiv \mathbf{x} \tag{2.5}$$

Depending on the underlying problem the inference steps f^t might take different forms, *e. g.* gradient descent steps or mean field updates, and they are usually fully defined by J and a few hyper parameters, *e. g.* by calculating the gradient of J and specifying the step size. We can now unroll the T inference steps and regard inference as one big prediction function F:

$$\hat{\mathbf{y}}^T = f^{T-1} \circ f^{T-2} \circ \cdots \circ f^0(\mathbf{y}^0) \equiv F(\mathbf{y}^0). \tag{2.6}$$

If we interpret F as a parameterized prediction function, we can ask how the parameters should be set in order to obtain optimal predictions. In image restoration, Tappen et al. (2007) were among the first to learn parameters of a Gaussian conditional random field (GCRF) by minimizing the loss between the posterior mode and the clean image. For a GCRF the inference function F is exact and entails solving a single linear equation system. In order to allow for sharp edges, weights for the filter responses in the GCRF are predicted from the noisy input image.

The shrinkage fields model of Schmidt and Roth (2014) is built upon the work of Tappen et al. (2007). They consider the FoE model (Roth and Black, 2011) which allows for efficient MAP inference by using half-quadratic splitting (HQS). Here, the posterior is augmented with auxiliary variables that decouple the arguments of the potential function in the prior from the filter responses of the experts. For inference, two steps are alternated. First, given the current estimate of the clean image the auxiliary variables are updated by a shrinkage function, corresponding to solving a one-dimensional optimization problem. Second, given the current set of auxiliary variables the posterior over the clean image is again a GCRF and its mode can be calculated efficiently in the Fourier domain when assuming circular boundary conditions of filter operations. Shrinkage fields now unroll this optimization and the shrinkage function is replaced by a trainable mapping which is parameterized as an ensemble of radial basis functionss (RBFs). Importantly, the parameters can be learned using loss based training. Interestingly, Schmidt and Roth found that the learned updates of auxiliary variables can not correspond to any potential function. Considering accuracy, the full shrinkage fields model outperforms its generative FoE baseline.

In a similar spirit to the cascades of shrinkage fields model, Chen and Pock (2017) propose to unroll inference in a diffusion model, which can also be interpreted as gradient descent iterations in a standard MAP estimation setting. They replace the gradient of the prior potential function with a flexible RBF model. In contrast to (Schmidt and Roth, 2014) the resulting algorithm, called trainable non-linear reaction diffusion (TNRD), is not restricted to Gaussian data terms and circular boundary conditions for convolutions. Hence, in addition to image denoising they also apply TNRD to JPEG deblocking and super-resolution, outperforming other discriminative and generative models at that time.

In two more recent works Lefkimmiatis (2017), 2018 extends the TNRD framework along two directions. First, he introduces box constraints that keep restored image intensities within a valid range. Second, he replaces the regularization of local filter responses by regularization of a non-local average of filter responses. Since all parameters of the model are trainable, he can easily extend his model to color image denoising by feeding more input channels and sharing the filter weights. In (Lefkimmiatis, 2018) he additionally removes the data term for AWGN denoising and replace it with a box constraint that require the denoised image to be within a σ-ball around the noisy image. He devices an inference scheme for this constrained energy based on a projected gradient descent approach. The parameters of the unrolled optimization are then learned from data. His method can adapt to different noise strengths by changing the threshold of the projection operator. During training he still needs to synthesize noisy

images with different noise levels to successfully learn parameters that work well across a broad range of noise levels.

MARRYING CNNS AND NON-LOCAL PROCESSING While the literature from the last section considered designing a differentiable network architecture by unrolling inference algorithms for classical denoising objective functions, in this section we look at neural network architectures that go beyond local convolutional and element-wise processing by integrating non-local processing layers. This is motivated by the success of model-based non-local approaches like BM3D (Dabov et al., 2006) or WNNM (Gu et al., 2014).

The recent works of Ahn and Cho (2017) and Yang and Sun (2018) closely resemble the BM3D algorithm. The basic idea is to replace BM3D's fixed *filter-and-threshold* pipeline for each matched patch group by a trainable neural network. The BM3D-Net of Yang and Sun (2018) directly predicts a denoised patch while the BMCNN of Ahn and Cho (2017) is trained to predict the noise residual. In addition, BMCNN computes the block matching on a pre-denoised version of the input image and augments the noisy patches with their pre-denoised counterparts. Consequently, while BM3D-Net can outperform the BM3D baseline, BMCNN can even compete with top-performing discriminative algorithms like DnCNN (Zhang et al., 2017a). However, it has to be noted that BMCNN uses DnCNN as a preprocessing step.

The recent NLNet (Lefkimmiatis, 2017) and UNLNet (Lefkimmiatis, 2018) both use collaborative filtering of patch groups as a trainable part in their unrolled inference approach. Here, block matching is conducted once on the noisy input patches and the indices of matched patches are used for all trained inference stages. Moreover, Lefkimmiatis (2018) conducted an oracle experiment, where he performs block matching on the noise-free ground truth image leading to an PSNR improvement of 0.7 dB on average. These promising results lead us to the idea of learning the feature space on which block matching is conducted instead of relying on the arbitrary choice of matching noisy patches. In Chapter 5 we show that this leads to significant improvements over competing non-local methods.

Liu et al. (2018) recently proposed a network that involves non-local processing by weighted averaging of neighboring pixels. The weights are determined based on the feature distance between pixels. Since the weighted average is differentiable they can obtain gradients with respect to the feature embedding. However their method is not able to apply a non-linear collaborative filtering like in BM3D or the other discussed approaches. Nevertheless, they can achieve an impressive improvement over the state of the art in many image restoration applications by repeatedly applying their non-local module with a stage-wise refinement of the weights used for averaging the pixel neighborhoods.

In an interesting recent work Cruz et al. (2018a) apply a non-local filter as post-processing to a pretrained denoising algorithm. Although their combined model could be trained end-to-end as long as the pretrained denoiser is differentiable, they do not consider this. Nevertheless, they achieve improvements over the baseline denoiser on images with a lot of recurrent structure. However, as shown in Chapter 5, their approach is not able to recover from error made by the initial denoising.

2.4 THEORETICAL AND PRACTICAL CONSIDERATIONS

In the following we discuss some theoretical and practical considerations regarding image denoising, namely bounds on the achievable denoising performance and how to deal with an unknown noise level and with noise that does not follow an *i. i. d.* Gaussian distribution.

BOUNDS ON DENOISING PERFORMANCE Given the vast zoo of denoising algorithms it is natural to ask, what optimal denoising accuracy we can expect. Several works have addressed this question in the past. Among the first were Chatterjee and Milanfar (2010). They derive a lower bound on the mean squared error (MSE) that depends on the noise strength and the covariance of clean patches. The tightness of the bound depends on how well the covariance can be estimated. Hence, they make the assumption that clean image patches cluster according to their photometric content. The derived bound is reported to be still quite far from the denoising accuracy of that time but it is unclear how much of this gap is to due to the bound being loose.

Levin and Nadler (2011) approached the problem of estimating bounds on the denoising accuracy in a non-parametric way. Specifically, they gather a large database of fixed-sized patches of natural images and study the mean squared error of reconstructing the center pixel from a noisy patch. This, as any other denoising algorithm, gives a lower bound on the achievable PSNR. At the same time, they consider the posterior variance of clean center pixels given the noisy observation, which they empirically show to be an upper bound on the denoising performance. In this framework, they evaluate the gap between the upper and lower bound as a function of the noise level of AWGN and the patch size. Their main findings can be summarized as follows: First, the stronger the noise the tighter the bounds even for larger support sizes, with state-of-the-art denoising algorithms being close to optimality. Second, for smaller noise levels the bounds are tight only for extremely small support sizes of the patches, *i. e.* less than 10 pixels, with the strongest denoisers being far away from the upper bound. This effectively means that for weak and medium noise, substantial gains can still be expected when the receptive field of a denoised pixel is large.

Building upon their previous work Levin et al. (2012) study the problem of denoising performance of non-parametric algorithms when increasing the size of patches. They find an analog to the curse of dimensionality, in that with scaling up the patch size it becomes increasingly hard to find similar patches in an external database. While this is not surprising, they also show that this problem is most severe for patches with high complexity, which is in line with the findings of Zontak and Irani (2011). Moreover, they derive the optimal denoising performance when assuming a simplified, piece-wise constant image model. Here, the optimal denoising performance follows a power-law dependent on the patch size which they empirically find to be consistent with the accuracy of a non-parametric method on finite patch sizes. They extrapolate these results and estimate that denoising performance might just increase by $0.5 - 1.0$ dB over BM3D.

The previously discussed efforts all need to make some assumptions on the noise distribution and the distribution of natural images. Thus, the conclusions drawn from these studies, albeit very valuable, need to be taken with a grain of salt concerning their applicability to real image denoising. Moreover, these studies define denoising accuracy in terms of the mean squared error of the denoised image. Since optimizing for the MSE leads to blurry results, recent research has also focused on optimizing the perceptual quality of restored images, *e. g.* in super resolution (Deng, 2018; Ledig et al., 2018; Sajjadi et al., 2017). These methods typically lead to sharper and thus more naturally looking results. However, the accuracy in terms of PSNR or MSE is worse. In an inspiring work Blau and Michaeli (2018) theoretically show that the two goals of optimizing accuracy and perceptual quality are at odds with each other. As long as an image restoration method only produces a single output, it has to trade off these two goals but can not achieve both at the same time.

BLIND DENOISING AND NOISE ESTIMATION Most algorithms that we presented in Sections 2.2 to 2.3 are *non-blind*, *i. e.* they assume knowledge of the noise distribution, *e. g.* the noise standard deviation in case of AWGN. In many practical applications, however, this information is not readily available. Here, so-called *blind* denoising algorithms come into play. There are two main approaches for blind denoising. First, data-driven models can be trained with noisy images having a large variety of noise levels as done, *e. g.*, for the DnCNN-B model (Zhang et al., 2017a) or for our model on raw image denoising in Chapter 5, where noise level functions used for training cover a broad range of realistic noise level functions. The second main approach to blind denoising comprises of a two step procedure where first the parameters of the noise distribution are estimated on the noisy image. The estimated noise characteristics are then supplied to a non-blind denoising algorithm. Since many denoising algorithms are very sensi-

tive to the right specification of the noise distribution, the accuracy of this two-step approach strongly depends on the accuracy of the noise estimation process.

A main hurdle in estimating the noise strength is the need to separate the local variance of the image signal from the noise variance. Hence, algorithms that look at the standard deviation of the noisy image, *e. g.* the early work of Meer et al. (1990), are bound to overestimate the noise variance. The algorithm of Rank et al. (1999) operates in the gradient domain, where image signals are sparser. They proceed by averaging local estimates of the noise variance. However, their estimate is still affected by the image signal. To overcome this problem, Liu et al. (2013) try to find patches with little texture from which they then estimate the noise variance by looking at the smallest eigenvalue of a principal components analysis (PCA) decomposition of those patches. However, selecting untextured patches is by itself a process that is adversely affected by noise. Hence, Liu et al. propose an iterative algorithm for refining the initial noise estimate. A similar approach is taken by Pyatykh et al. (2013). Chen et al. (2015b) pointed out a fundamental bias in the methods of Liu et al. (2013) and Pyatykh et al. (2013). They show that the smallest eigenvalue systematically underestimates the noise variance since the patch covariance matrix is based on a finite set of sampled patches. They propose to use a robust estimator that selects the mean of the eigenvalues after robustly removing large outliers that stem from the image signal. The resulting estimator for the noise variance is shown to be more accurate in terms of bias and variance. Zoran and Weiss (2009) observed that the kurtosis of natural images is roughly constant across scales while it sharply falls off for white noise. They exploit this behavior for estimating the noise variance of noisy images corrupted with AWGN.

When dealing with Poisson-Gaussian noise, as in our work on capturing realistic ground truth data in Chapter 3, the noise strength is often assumed to be a linear function of the image signal. Hence, there are now two parameters to be estimated, *i. e.* the slope and the offset of this linear noise level function. Foi et al. (2008) propose to decompose the noisy image into sets of pixels with similar intensity and compute the mean intensity and the variance for each set. These mean-variance pairs are then used to robustly estimate the linear relation between intensity and noise variance. Mäkitalo and Foi (2014) refine this approach by additionally minimizing the non-Gaussianity of the output of the VST transformed image signal. In (Liu et al., 2014) a generalized intensity dependent model is fitted to mean-variance pairs that are obtained from low rank patch groups. The Noise Clinic (Lebrun et al., 2014) employs a carefully tuned algorithm for signal dependent noise estimation which is integrated into a multiscale denoising approach.

The noise distribution gets even more complicated for images that undergo the non-linear and often unknown camera processing pipeline (Karaimer and Brown, 2016). Liu et al. (2008) propose to model the noise distribution as a generalized signal dependent Gaussian. The noise level function for each channel is parameterized as a weighted combination of basis functions that were obtained by a PCA decomposition of measured noise level functions. The parameters are fitted to the mean-variance pairs measured on the noisy images, taking into account that the variance is typically overestimated. They show the applicability of their method both on simulated and real data. However, their approach does not take into account the chromatic correlations of noise that are introduced, *e. g.*, by the demosaicing or gamut mapping. This is addressed by Nam et al. (2016) who parameterize the noise distribution as a signal dependent three-dimensional Gaussian distribution. To deal with the high-dimensional parameter space they train an MLP regressor that predicts the noise covariance matrix from an input patch. However, they need to train a specific prediction model for each combination of camera and ISO level, hindering immediate practical applicability. In a recent work, Chen et al. (2018b) go even one step further and do not assume any parametric model of the noise distribution at all. Instead they try to extract noise patterns from sufficiently constant image regions and then train a GAN to learn the noise distribution. As a consequence, they can query the noise distribution by sampling from it, which they exploit to synthesize training data for a discriminative denoising model, *e. g.* DnCNN (Zhang et al., 2017a). Despite having an underlying restriction of only modeling additive noise, the resulting blind denoiser shows very good accuracy on our realistic benchmark dataset, *cf.* Chapter 3.

A notable third alternative to blind denoising is given by treating the unknown noise level in a Bayesian framework. For example the BLS-GSM algorithm (Portilla, 2004) tries to maximize the posterior of the clean image and the noise variance given the noisy image in an iterative fashion akin to expectation maximization (EM). Being still a bit more Bayesian, Schmidt et al. (2011) put a prior on the noise variance and obtain the posterior over the clean image while marginalizing over the noise variance.

ASSESSING DENOISING ACCURACY We also want to briefly discuss the problem of assessing the denoising accuracy. When the ground truth clean image is available, we can look at so-called *full-reference* quality metrics that compare the denoised result $\hat{\mathbf{y}}$ to the ground truth image $\mathbf{y}$. The PSNR and the SSIM index (Wang et al., 2004) that we report in this thesis are two popular examples of full-reference metrics. More sophisticated methods include information theoretic approaches like IFC (Sheikh et al., 2005) and VIF (Sheikh and Bovik, 2006) which measure the mutual information between the distributions

of wavelet coefficients coming from the denoised and ground truth images, respectively.

In the absence of ground truth, experimental evaluation often resorts to visual inspection of the denoised results. However, this approach is subjective when only few observers rate the denoised image, while a user study with many participants is costly and time consuming. Hence, several *no-reference* quality metrics have been proposed that aim at judging the perceptual quality from the denoised image alone, thus taking the role of an objective observer. The DIIVINE (Moorthy and Bovik, 2011), BLINDS-II (Saad et al., 2012) and BRISQUE (Mittal et al., 2012) measures discriminate features from the distributions of local image statistics from either clean images or distorted images. The features are derived by fitting parametric models such as generalized Gaussians to the empirical distribution of filter responses. From these features, a regression function predicts a quality score that should align well with human judgments given by mean opinion scores. Recently, Ma et al. (2017a) propose to train a random forest model to directly regress a perceptual quality index derived from human ratings.

When considering the special task of removing white Gaussian noise, an alternative way to estimate the image quality is given by Stein's unbiased risk estimator (SURE) (Stein, 1981). Assuming knowledge of the exact noise strength and continuity of the denoising function, SURE allows to obtain an unbiased estimate of the MSE of the denoising result without knowing the ground truth. This has been used in an ensemble method to weight contributions of weak denoisers (Blu and Luisier, 2007; Luisier and Blu, 2008) or to tune hyper parameters of the nonlocal means denoiser (Van De Ville and Kocher, 2009) and more general black box denoising algorithms (Ramani et al., 2008). The SURE estimator can also be applied for Poisson-Gaussian noise (Luisier et al., 2011). In two interesting recent works, Soltanayev and Chun (2018) and Metzler et al. (2018) show that the SURE principle can also be used to train denoising networks in an unsupervised fashion, just by assuming that the noise distribution is Gaussian with zero mean. Interestingly, denoisers trained in this way are very competitive with their counterparts trained in a supervised way. This line of work might seem to be of little relevance for denoising natural images taken with a digital camera, since we can always capture virtually noise-free images and synthesize training data with the same Gaussian noise assumption from these. However, unsupervised training becomes intriguing whenever it is hard to obtain clean images in the first place. For the example of X-ray images capturing a virtually noise-free image requires a dose of radiation that would be damaging to the imaged subject.

2.5 DENOISING FOR REGULARIZING INVERSE PROBLEMS

Finally, we want to highlight that denoising is not just useful to increase the perceptual quality of images. Therefore, we discuss a recent stream of work that employs image denoising to regularize more general image restoration problems.

It has been recognized that traditional optimization procedures based on variable splitting, *e. g.* alternating direction method of multipliers (ADMM) (Boyd et al., 2011; Gabay and Mercier, 1976; Glowinski and Marroco, 1975) or HQS, contain steps that can be interpreted as an instance of the image denoising problem. In their seminal work, Venkatakrishnan et al. (2013) relate the prior step in the ADMM framework to the solution of a regularized Gaussian denoising problem. They show that any denoising algorithm can be plugged in instead of the original update step. As an important consequence – that is also shared with the methods discussed in the following – the prior term does not need to be specified explicitly but is defined implicitly by the choice of the used denoiser. This allows to easily encode a "self-similarity" prior by using non-local denoising algorithms like BM3D (Dabov et al., 2006). The plug-and-play method has been extended to primal-dual ADMM by Ono (2017). Their proposed algorithm also allows to include hard constraints such as non-negativity of the restored image intensities.

In a similar spirit Heide et al. (2014) show that Gaussian denoising can be related to proximal operators (Parikh and Boyd, 2014). Consequently, they swap the prior proximal operator in the primal-dual optimization framework (Chambolle and Pock, 2011) with a combined operator consisting of the BM3D denoiser (Dabov et al., 2006) that implicitly encodes a self-similarity prior, a total variation (TV) prior and a gradient consistency prior. The resulting method is applied to several image restoration problems, including demosaicing, burst denoising and HDR reconstruction.

The prior proximal step in the HQS framework can also be cast as a denoising problem as shown by Zhang et al. (2017b). They propose to train a set of CNN denoisers, each tuned to a specific noise level. The proximal step is then conducted by the denoiser whose noise level corresponds best to the continuity strength of the proximal operator. Recently, Xiao et al. (2018) propose to train the denoising networks for the proximal step discriminatively instead of using an off-the-shelf denoiser. Since the prior proximal operator is not task specific, they learn their pipeline jointly for multiple image restoration problems, thus elegantly combining the modularity of generative approaches to image restoration with the strength of discriminative learning.

Besides these works that integrate denoising in variable splitting frameworks, we want to highlight two other lines of work that employ denoising as subroutine for general image restoration. First, it was

noted by Alain and Bengio (2014) that the reconstruction error of an optimal denoising auto-encoder corresponds to the gradient of the data distribution smoothed with the noise level used for generating the training examples. As shown by Bigdeli et al. (2017) this observation can be harnessed for general low-level image restoration problems in a gradient descent framework for MAP estimation. Specifically, the prior is not modeled explicitly but its gradient is given by the reconstruction error of a pre-trained denoising auto-encoder. In a second work, Romano et al. (2017) make the denoising function a first-class citizen by specifying a prior term based on the residual between an image and its denoised version. The prior favors images which either have a small residual or where the residual is uncorrelated to the image.

Part I

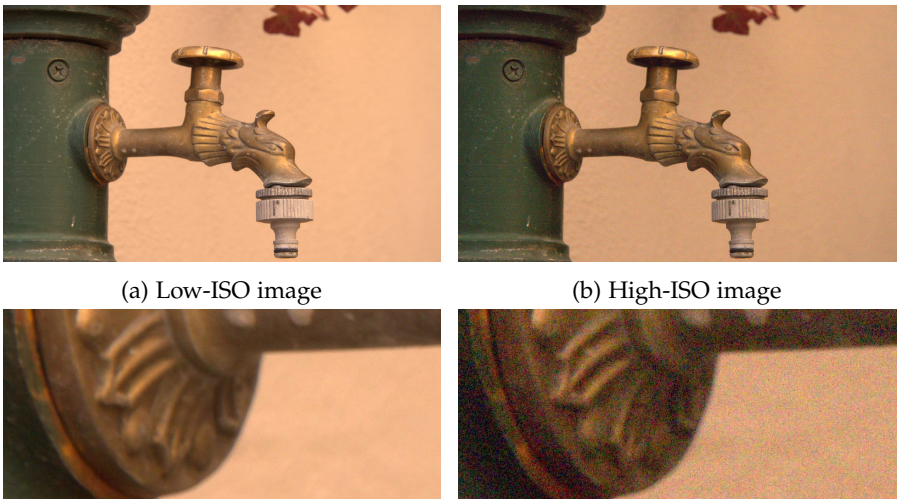
BENCHMARKING METHODOLOGY

3

AN IMAGE DENOISING BENCHMARK USING REAL IMAGES

CONTENTS

Lacking realistic ground truth data, image denoising techniques are traditionally evaluated on images corrupted by synthesized *i. i. d.* Gaussian noise. We aim to obviate this unrealistic setting by developing a methodology for benchmarking denoising techniques on real photographs. We capture pairs of images with different ISO values and appropriately adjusted exposure times, where the nearly noise-free low-ISO image serves as reference. To derive the ground truth, careful post-processing is needed. We correct spatial misalignment, cope with inaccuracies in the exposure parameters through a linear intensity transform based on a heteroscedastic Tobit regression model, and remove residual low-frequency bias that stems, *e. g.*, from minor illumination changes. We then capture a novel benchmark dataset, the *Darmstadt Noise Dataset (DND)*, with consumer cameras of differing sensor sizes. One interesting finding is that various recent techniques that perform well on synthetic noise are clearly outperformed by BM3D on photographs with real noise. Our benchmark delineates realistic evaluation scenarios that deviate strongly from those commonly used in the scientific literature. This chapter is based on (Plötz and Roth, 2017) and extends our prior work by providing an analysis of the noise distribution in Section 3.3 and by reviewing the usage and adoption of the benchmark in Section 3.8.



(a) Low-ISO image (b) High-ISO image

(c) Zoom-in of the low-ISO image (left) and the high-ISO image (right)

Figure 3.1. An image pair of a nearly noise-free low-ISO and a noisy high-ISO image from our dataset. Note, that we work with RAW images and show JPEGs for better display.

3.1 INTRODUCTION

Noise is inherent to every imaging system. Especially in low-light scenarios, it often severely degrades the image. Therefore, a large variety of denoising algorithms have been developed to deal with image noise, *e. g.* (Buades et al., 2005b; Burger et al., 2012; Chen et al., 2015c; Dabov et al., 2007; Dong et al., 2013; Mairal et al., 2009; Portilla et al., 2003; Roth and Black, 2009; Zoran and Weiss, 2011). Even though images with real sensor noise can be captured easily, it is much less straightforward to know what the true noise-free image should be. Thus, the quantitative evaluation of denoising methods by and large relies on adding synthetic *i. i. d.* Gaussian noise to mostly clean images, *e. g.* (Jancsary et al., 2012; Portilla et al., 2003; Roth and Black, 2009). Photographs with real noise are at best used for a qualitative analysis (Mairal et al., 2009; Nam et al., 2016), but often not at all. This is quite problematic, since noise in real photographs is not *i. i. d.* Gaussian (Foi et al., 2008; Liu et al., 2008), yet even seemingly minor details of the synthetic noise process, such as whether the noisy values are rounded to integers, can have a significant effect on the relative performance of methods (Chen et al., 2015c; Schmidt and Roth, 2014).

The goal of this chapter is to address these challenges by developing a methodology for benchmarking denoising algorithms by means of real photographs. At its core is the simple idea of capturing pairs of noisy and almost noise-free images by imaging the same scene from the same viewpoint with different analog gains (ISO values), see Fig. 3.1. By inversely adjusting the exposure time, the underlying noise-free image intensities should theoretically stay constant. In practice, we observe various causes for changing image intensities, prohibiting

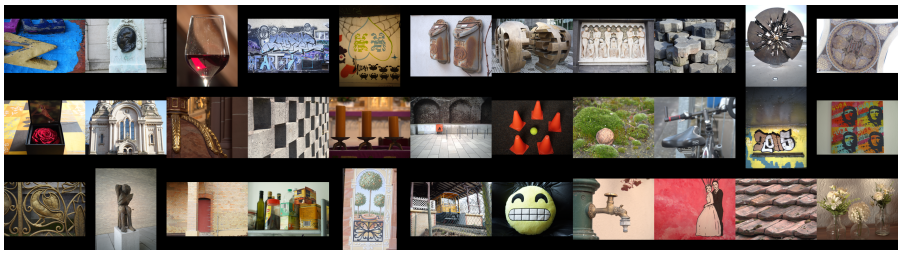


Figure 3.2. An overview of the scenes used in our benchmark dataset (subset shown).

the direct use of the low-ISO image as ground truth. Since these scene variations are non-trivial, we contribute a careful *post-processing procedure* that takes into account the statistical properties of the image formation process. As part of this pipeline we propose a *heteroscedastic Tobit regression model* generalizing the work of Tobin (1958), which allows to remove linear dependencies between the intensities of both images that arise as neither the analog gain of the sensor nor the exposure time can be controlled completely accurately in practice. Our model faithfully accounts for clipping as well as signal-dependent noise, which is crucial as shown experimentally. Furthermore, minimal changes in the illumination can lead to a low-frequency bias, which we remove by high-pass filtering the residual between noisy and reference image in a transformed domain in which the noise process is zero-mean. Lastly, moving objects and minuscule camera shake between exposures are treated by manual annotation and simple Lucas-Kanade subpixel alignment (Lucas and Kanade, 1981), respectively.

Based on this acquisition pipeline, we capture a *real-world dataset* of image noise, called Darmstadt Noise Dataset (DND). We use four consumer cameras, ranging from a smartphone with a 1/2.3 inch sensor to a full-frame interchangeable lens camera. Images are taken across a wide range of different ISO values in realistic photographic situations, providing a novel reference dataset for benchmarking denoising algorithms in realistic conditions. Our dataset consists of 50 scenes and is publicly available.[1] Figure 3.2 shows a subset of the scenes.

Our realistic dataset enables *interesting insights* into the performance of recent denoising algorithms. We find that a number of current techniques (*e. g.*, NCSR (Dong et al., 2013), WNMM (Gu et al., 2014), TNRD (Chen et al., 2015c)) that – based on previous analyses with synthetic *i. i. d.* Gaussian noise – were presumed to outperform the by now classic BM3D (Dabov et al., 2007), do in fact perform worse than BM3D on photographs with real noise. Moreover, our analysis reveals that noise strengths for consumer cameras are significantly lower than what is usually assumed in the scientific literature when evaluating denoising algorithms. We further highlight the importance of applying denoising before the non-linear camera processing pipeline (Park et al.,

1 https://noise.visinf.tu-darmstadt.de

2009). Our findings strongly question the *practical relevance* of previous synthetic evaluation methodologies.

3.2 RELATED WORK

Since noise is abundant in any imaging system, its statistical properties have been well studied. Thorough analyses have been provided for CCD (Healy and Kondepudy, 1994) and CMOS image sensors (El Gamal and Eltoukhy, 2005). One inevitable source of noise is induced by the stochastic arrival process of photons hitting the sensor – so-called shot noise. Since it follows a Poisson distribution, its variance is proportional to the mean intensity at a specific pixel and is hence not stationary across the whole image. Other noise sources originate from the electronics within the sensor chip and from discretization (El Gamal and Eltoukhy, 2005; Foi et al., 2008; Healy and Kondepudy, 1994).

Although the image noise variance depends on the underlying intensity, the majority of denoising algorithms ignore this and evaluates against artificial, stationary noise, usually assumed *i. i. d.* Gaussian, (*e. g.* Portilla et al., 2003; Roth and Black, 2009; Zoran and Weiss, 2011). Other works specifically aim to *model intensity-dependent noise* (Liu et al., 2008; Luisier et al., 2011). The main idea there is to model the noise distribution as a heteroscedastic Gaussian, whose variance is intensity-dependent. This is valid since the Poissonian components of the total noise can be approximated well with a Gaussian. Other approaches first apply a variance stabilizing transform (Foi, 2009; Mäkitalo and Foi, 2013) and subsequently employ a denoising method for stationary Gaussian noise. However, the transform may make the noise distribution non-Gaussian (Zhang et al., 2015).

There have been attempts to validate denoising algorithms on real data at a small scale (Liu et al., 2014; Zhu et al., 2016). They rely on recovering a noise-free image by temporal averaging several noisy observations. However, they ignore the fact that the noise process is not zero-mean due to clipping effects (Foi, 2009), whereas we show that it is important to consider this bias when creating a denoising ground truth. They also do not take potentially further non-linear processing of raw intensities (Karaimer and Brown, 2016) into account.

To the best of our knowledge, the only prior effort on benchmarking denoising with real images is the RENOIR dataset (Anaya and Barbu, 2018). It also relies on taking sets of images of a static scene with different ISO values, but the post-processing is less refined. Image pairs appear to exhibit spatial misalignment, the intensity transform does not model heteroscedastic noise, and low-frequency bias is not removed. Our experiments indicate that ignoring these sources of error significantly affects the realism of the dataset. Moreover, the work of

Anaya and Barbu (2018) is based on 8 bit demosaiced images while we work with untainted linear raw intensities.

Due to the restricted size of our benchmark, we exclusively provide a testing set. However, subsequent datasets on realistic image denoising increase the number of images significantly and hence also provide training sets on which discriminative denoisers can be fitted. The SIDD dataset (Abdelhamed et al., 2018) comprises of 30 000 images associated to 10 scenes, each pictured with 5 different smartphone cameras and 4 different lighting conditions. Thus, 150 images are captured per setting, allowing Abdelhamed et al. to use a more refined post-processing pipeline than ours, entailing locally adaptive spatial alignment, outlier removal, and an identification of defective pixels. The SID dataset (Chen et al., 2018a) provides a dataset of roughly 5000 short exposure images that were taken under extreme low light while the accompanying reference image were taken with a much longer exposure. The benchmark allows to evaluate the task of recovering the developed long exposure sRGB image from the raw short exposure image, which not only requires denoising but also faithfully modeling the camera processing pipeline.

It is often useful to measure the noise characteristics of a sensor at a certain ISO level. The European Machine Vision Association (2012) proposes to illuminate the sensor with approximately constant irradiation and subsequently aggregate intensity measurements *spatially*. This is repeated for different irradiation levels to capture the intensity dependence of the noise. Foi et al. (2007) and Moldovan et al. (2006) propose a less tedious capture protocol similar to ours, where multiple exposures of a static scene are used to aggregate the measurements at every pixel site *temporally*. In contrast, our Tobit regression allows to estimate the parameters of the noise process by having access to just two images.

3.3 A MODEL OF IMAGE SENSOR NOISE.

Let us first review models that describe the statistical characteristics of the noisy image signal in unclipped digital images (European Machine Vision Association, 2012; Healy and Kondepudy, 1994). Although these models are simplified in the sense that they neglect some noise sources like fixed-pattern noise or defective pixels, they will allow us to understand the dependence of image noise on the camera's analog gain K (*i. e.* its ISO value) and exposure time τ. The observed intensity of a pixel N_I can be written as (Healy and Kondepudy, 1994)

$$N_I = K \cdot (N_e + N_d + N_{o_1}) + N_{o_2} + N_q, \tag{3.1}$$

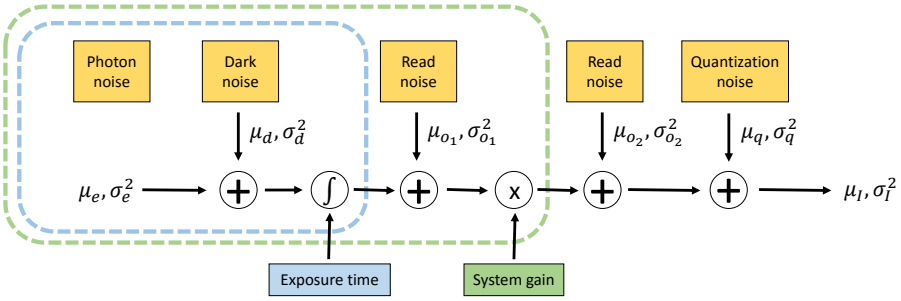


Figure 3.3. Schematic model of the aggregation of different noise sources in the imaging process. Parts within the blue box are influenced by the exposure time, parts in the green box by the gain

where the noise constituents are distributed as

$$N_e \sim \mathcal{P}(\mu_e) = \mathcal{P}(\eta \cdot F_p \cdot \tau) \tag{3.2}$$
$$N_d \sim \mathcal{P}(\mu_\mathrm{d}) = \mathcal{P}(F_d \cdot \tau) \tag{3.3}$$
$$N_{o_1} \sim \mathcal{N}(0, \sigma_{o_1}^2) \tag{3.4}$$
$$N_{o_2} \sim \mathcal{N}(0, \sigma_{o_2}^2) \tag{3.5}$$
$$N_q \sim \mathcal{U}(-0.5\Delta_e, 0.5\Delta_e). \tag{3.6}$$

The number of electrons[2] N_e caused by the incident light is Poissonian distributed with mean intensity μ_e, which depends on the quantum efficiency η of the sensor, the incoming photon flux F_p at the pixel site, and the exposure time. The electrons due to dark current N_d also follow a Poisson distribution with its mean depending on the generation rate of dark electrons F_d and the exposure time. The variances σ_e^2 and σ_d^2 of N_e and N_d are equal to their mean intensities, respectively. The sensor read-out causes noise that can be described by Gaussian random variables, where N_{o_1} is the read noise before and N_{o_2} is the read noise after amplification. We assume that any non-zero bias in these variables (*e. g.*, fixed-pattern noise) is corrected for by the camera sensor. Finally, the signal is quantized, which can be interpreted as adding uniformly distributed noise N_q with variance σ_q^2 to the signal (Oppenheim and Schafer, 1975). N_q takes values in the range $[-0.5\Delta_e, 0.5\Delta_e]$ where Δ_e is the quantization step size. See Fig. 3.3 for a visualization of this standard model of the imaging process and the noise sources that influence the final raw pixel output. For sufficiently large incoming photon flux the Poisson distribution can be well approximated by a Gaussian. Moreover, if the image signal is large compared to the quantization width we can approximate the whole distribution of an unclipped noisy pixel intensity as Gaussian.

We now study how the mean and variance of the observed signal depend on the gain K and the exposure time τ. Due to the linearity of

2 Since charge and voltage are convertible we do not distinguish between both.

the expectation the mean signal as a function of gain and exposure time is given by

$$\begin{aligned}\mu_I(K,\tau) &= K\cdot(\mu_e+\mu_\mathrm{d}+\mu_{o_1})+\mu_{o_2}+\mu_q && \text{(3.7a)}\\ &= K\cdot\tau\left(\eta\cdot F_p+F_d\right)+K\cdot\mu_{o_1}+\mu_{o_2}+\mu_q && \text{(3.7b)}\\ &= K\cdot\tau\left(\eta\cdot F_p+F_d\right), && \text{(3.7c)}\end{aligned}$$

where the last equality follows from N_{o_1}, N_{o_2} and N_q having zero mean. It is also reasonable to assume that all noise sources are independent and hence the total variance of the signal as a function of gain and exposure time is given by

$$\begin{aligned}\sigma_I^2(K,\tau) &= K^2\cdot\left(\sigma_e^2+\sigma_\mathrm{d}^2+\sigma_{o_1}^2\right)+\sigma_{o_2}^2+\sigma_q^2 && \text{(3.8a)}\\ &= K^2\cdot\left(\mu_e+\mu_\mathrm{d}+\sigma_{o_1}^2\right)+\sigma_{o_2}^2+\sigma_q^2 && \text{(3.8b)}\\ &= K^2\tau\cdot\left(\eta\cdot F_p+F_d\right)+K^2\cdot\sigma_{o_1}^2+\sigma_{o_2}^2+\sigma_q^2. && \text{(3.8c)}\end{aligned}$$

We can observe that the variance is linear in the unamplified image signal and that the gain controls the slope and offset of this linear relation ship.

The main idea behind our capture protocol will be to obtain images with varying noise levels but comparable overall intensity by multiplying the gain K by some factor n and at the same dividing the exposure time τ by the same factor. Given Eq. (3.7c), we can readily verify that the mean intensity is unchanged:

$$\mu_I(nK,\tau/n)=nK\cdot\frac{\tau}{n}\left(\eta\cdot F_p+F_d\right)=\mu_I(K,\tau). \tag{3.9}$$

The main reason for this result is that all components of the signal with non-zero mean depend proportionally on the gain and inversely proportional on the exposure time. For the variance, on the other hand, we obtain

$$\begin{aligned}\sigma_I^2(nK,\tau/n) &= n^2K^2\tau/n\cdot\left(\eta\cdot F_p+F_d\right)+n^2K^2\cdot\sigma_{o_1}^2+\sigma_{o_2}^2+\sigma_q^2 && \text{(3.10a)}\\ &= nK^2\tau\cdot\left(\eta\cdot F_p+F_d\right)+n^2K^2\cdot\sigma_{o_1}^2+\sigma_{o_2}^2+\sigma_q^2 && \text{(3.10b)}\\ &= n\cdot\sigma_I^2(K,\tau)+\Big(n-1\Big)\Big(nK^2\cdot\sigma_{o_1}^2-\sigma_{o_2}^2-\sigma_q^2\Big). && \text{(3.10c)}\end{aligned}$$

This result can be interpreted as follows: For pixels with high irradiance the noise is dominated by shot noise and hence $\sigma_I^2(nK,\tau/n)\approx n\cdot\sigma_I^2(K,\tau)$. In this regime, the variance scales linearly with the gain. The less the irradiance becomes, the higher the weight of the noise before amplification $\sigma_{o_1}^2$ will become and hence $\sigma_I^2(nK,\tau/n)\approx n^2\cdot\sigma_{o_1}^2(K,\tau)$. Note that this model only holds for unclipped pixels. In practice, however, under- and overexposed pixels become clipped. The mean and variance of the clipped signal will deviate from the mean and vari-

ance of the unclipped signal. We address this in our noise estimation procedure (Sec. 3.5).

3.4 MODEL OF CLIPPED IMAGES AND DATA ACQUISITION

Let us now turn to a model of actual clipped noisy images as recorded by our capture protocol that was carried out to acquire our benchmark dataset. Figure 3.4 summarizes the capture protocol as well as steps taken during post-processing.

IMAGE FORMATION. Capturing a noisy image x_n can be described by adding noise to a latent noise-free image y_n and afterwards clipping the intensities to account for the saturation of pixels on the sensor:

$$x_n = \mathrm{clip}\big(y_n + \epsilon_n(y_n)\big), \tag{3.11}$$

where $\mathrm{clip}(y) = \min(\max(y, 0), 1)$. Given the analysis in Section 3.3 we can identify y_n by the expectation of the unclipped noisy signal (Eq. 3.7c), and ϵ_n can be modeled as Poisson-Gaussian noise whose strength depends on the noise-free intensity. Following (Azzari and Foi, 2014; Foi, 2009) and the result of (Eq. 3.8c), we approximate the noise distribution with a heteroscedastic Gaussian

$$\epsilon_n(y_n) \sim \mathcal{N}\big(0, \sigma_n(y_n)\big) \tag{3.12}$$

$$\text{with} \quad \sigma_n^2(y_n) = \beta_1^n y_n + \beta_2^n, \tag{3.13}$$

where $\sigma_n(y_n)$ is called the *noise level function* with parameters $\boldsymbol{\beta}^n$. Due to the clipping, naïve temporal or spatial averaging of the noisy observations will yield a bias, *i. e.* $\mathbb{E}\left[x_n \mid y_n\right] \neq y_n$. However, we can express $\mathbb{E}\left[x_n \mid y_n\right]$ analytically in terms of y_n and $\sigma_n(y_n)$, see (Foi, 2009) for details, and denote this relation as

$$\mathcal{A}(y_n) \doteq \mathbb{E}\left[x_n \mid y_n\right]. \tag{3.14}$$

Ideally, we would want to use y_n as ground truth for denoising x_n. However, since y_n is not available, we propose to take another picture x_r that shows the same scene as x_n, but is affected only little by noise. Since the parameters $\boldsymbol{\beta}$ of the noise-level-function depend mainly on the camera sensor and on the ISO value (Eq. 3.8c, European Machine Vision Association (2012)), we achieve this by using a low ISO value to obtain the reference image x_r.

CAPTURE PROTOCOL AND RESIDUAL ERRORS. As this reference image x_r is captured at a different time instant and with a different exposure time and ISO value than x_n, it is generated from a second latent image y_r with noise parameters $\boldsymbol{\beta}^r$, analogously to Eq. (3.11). In practice, we take the reference at the base ISO level of the camera, while

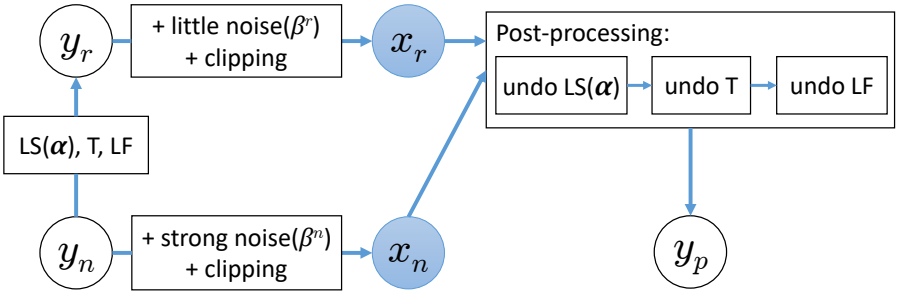


Figure 3.4. Image formation process underlying the observed low-ISO image x_r and high-ISO image x_n. They are generated from latent noise-free images y_r and y_n, respectively, which in turn are related by a linear scaling of image intensities (LS), a small camera translation (T), and a residual low-frequency pattern (LF). To obtain the denoising ground truth y_p, we apply post-processing to x_r aiming at undoing these undesirable transformations.

the ISO value for the noisy image is n times larger. To compensate this, the reference image is taken with n times the exposure time. All other camera parameters including aperture, white balance, and focus remain constant. Since the latent, noise-free image intensity is proportional to both the ISO value and the exposure time, *in theory* our capture protocol leaves the noise-free image intensities invariant, *i. e.* $y_n = y_r$, (*cf.* Eq. 3.9). As x_r exhibits only very little noise, *i. e.* $x_r \approx y_r$, we could use x_r instead of y_n as denoising ground truth.

For the noise-free intensity to truly stay the same, the captured scene and the camera have to be static and the illumination has to remain constant. Neither is generally the case. To minimize the effect of camera shake and scene variation during acquisition, we developed an Android app that quickly issues all necessary commands to the camera over WiFi. We mount the camera on a sturdy tripod with a stabilizing weight attached. Moreover, we use mirrorless cameras, which reduces vibrations due to mirror flapping compared to DSLRs. Despite this careful protocol, we still observe residual errors that we undo using the pipeline detailed in Section 3.5; post-processing x_r results in a new image y_p. In Section 3.6 we show that y_p is now sufficiently close to y_n and hence use y_p as ground truth for our benchmark.

Table 3.1. Cameras used for capturing the DND dataset.

Camera	# img.	Sensor size [mm]	Res. [Mpix]	ISO
Sony A7R	13	36×24	36.3	100 – 25.6k
Olympus E-M10	13	17.3×13	16.1	200 – 25.6k
Sony RX100 IV	12	13.2×8.8	20.1	125 – 8k
Huawei Nexus 6P	12	6.17×4.55	12.3	100 – 6.4k

Figure 3.5. Manually annotated binary mask image used for the above post-processing. Red pixels are not considered during post-processing.

FURTHER DETAILS. For our image database described in Section 3.7 we use four different cameras, see Table 3.1. The cameras span a substantial range of sensor sizes from $1/2.3$ inch to a full-frame sensor. We extract linear raw intensities from the captured images using the free software *dcraw*. Afterwards we scale image intensities to fall inside the range $[0, 1]$ by normalizing with the black and white level.

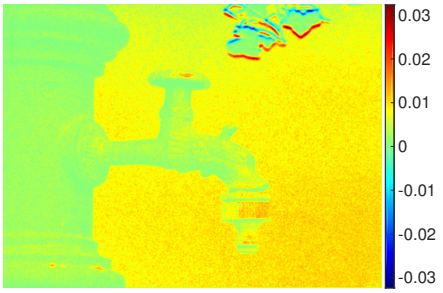
3.5 POST-PROCESSING

Our post-processing aims at undoing undesirable transformations between the latent images y_n and y_r. These are revealed by looking at the difference images between the low-ISO image x_r and high-ISO image x_n (Fig. 3.6a). Specifically we consider the *debiased residual image* $R(x_r)$ with

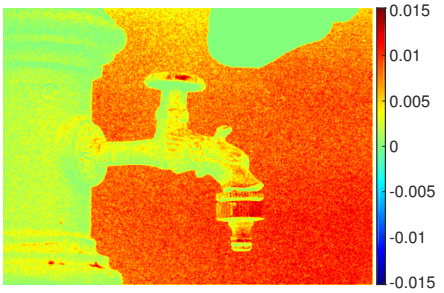
$$R(\cdot) \doteq \mathcal{A}(\cdot) - x_n. \tag{3.15}$$

From Eq. (3.14) it immediately follows that the *ground truth debiased residual image* $R(y_n)$ is zero-mean. However, from Fig. 3.6a it is apparent that $R(x_r)$ is not zero-mean. We trace this to four sources of errors that need to be corrected for in order to relate the intensities of a certain pixel across the different exposures: *(i)* In general scenes individual objects may move during the capture procedure; *(ii)* spatial sub-pixel misalignments may be caused by small camera vibrations, *e. g.*, due to the mechanical shutter; *(iii)* the lighting of the scene may change slightly during capture, outdoors for example because of moving clouds, indoors for example due to light flicker; *(iv)* linear intensity changes arise from the fact that neither the analog gain nor the exposure time can be perfectly controlled. Note, that the severity of *(i)–(iii)* aggravates the more pictures are taken, thus complicating the use of temporal averaging methods for creating denoising ground truth in realistic scenes. Our capture protocol strikes a balance between *(i)–(iii)* and *(iv)* by requiring the minimum of only two exposures, while creating the need to account for linear intensity changes.

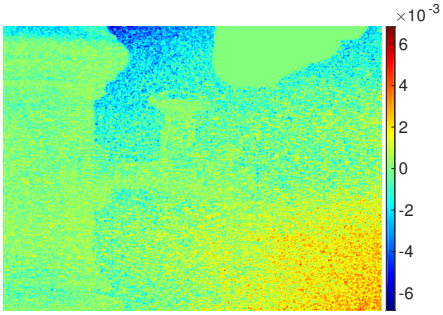
We need to cope with these four sources of errors to obtain an accurate ground truth. We address *(i)* by masking objects with a simple



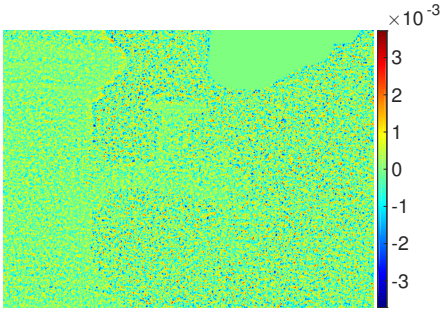
(a) Difference (residual) image $R(x_r)$ between Figs. 3.1a and 3.1b



(b) After manually masking moving objects



(c) Residual $R(y_p'')$ after linear intensity scaling and alignment



(d) Final residual $R(y_p)$ after subsequent removal of low-frequency residuals

Figure 3.6. Difference between blue channels of low- and high-ISO images from Fig. 3.1 after various post-processing stages. Images are smoothed *for display* to highlight structured residuals, attenuating the noise.

GUI tool. Our post-processing aims at undoing *(ii)–(iv)*, *cf.* Fig. 3.4. We model *(ii)* as a global 2D translation and *(iv)* as a linear scaling of pixel intensities, both of which can be inverted given an estimate of their underlying parameters. Any remaining low-frequency bias *(iii)* is removed in a final filtering step, producing the post-processed image y_p. We now detail these steps.

MASKING MOVING OBJECTS. When taking images outside a lab environment, in general we can not avoid moving objects in-between capturing the noisy and reference image, *e. g.* the red leaf that is slightly shifted between the two exposures of Figs. 3.1a and 3.1b. To exclude these moving objects from the rest of our post-processing, we manually annotate moving objects with a simple GUI, thus creating a binary mask. An example mask is shown in Fig. 3.5. When using the noisy and post-processed reference images as part of our benchmark we will also exclude masked pixels from the evaluation by not sampling crops that contain any masked pixel, *cf.* Section 3.7.

LINEAR INTENSITY CHANGES. Changing the analog amplifier gain and the exposure time introduces a linear relationship between y_n and y_r (Fig. 3.6b), since neither of those parameters can be controlled with perfect accuracy:

$$y_n = \alpha(y_r) = \alpha_1 y_r + \alpha_2, \tag{3.16}$$

where the offset α_2 accounts for inaccuracies of the recorded black level. As we do not have access to y_n and y_r, we need to estimate α_1, α_2 from the observed images. From Eq. (3.11), we relate x_r and x_n as

$$\begin{aligned} x_n &= \operatorname{clip}\big(y_n + \epsilon_n(y_n)\big) && (3.17)\\ &= \operatorname{clip}\big(\alpha(y_r) + \epsilon_n(\alpha(y_r))\big) && (3.18)\\ &\overset{*}{=} \operatorname{clip}\big(\alpha(x_r + \epsilon_r(y_r)) + \epsilon_n(\alpha(y_r))\big) && (3.19)\\ &\approx \operatorname{clip}\big(\alpha(x_r + \epsilon_r(x_r)) + \epsilon_n(\alpha(x_r))\big) && (3.20)\\ &= \operatorname{clip}\big(\alpha(x_r) + \alpha_1\epsilon_r(x_r) + \epsilon_n(\alpha(x_r))\big). && (3.21) \end{aligned}$$

The equality denoted with $*$ holds for non-clipped pixels in x_r, which are easily identified. The approximation defines the noise distributions in terms of the observed x_r instead of the unknown intensities y_r, since x_r is affected only little by noise. Exploiting that our capture protocol ensures that α_1 and α_2 are very close to 1 and 0, respectively, we can further approximate the scaled noise $\alpha_1\epsilon_r(x_r)$ as the noise of the linearly transformed image $\alpha(x_r)$:

$$\begin{aligned} \alpha_1\epsilon_r(x_r) &\sim \mathcal{N}\big(0, \alpha_1\sqrt{\beta_1^r x_r + \beta_2^r}\big) && (3.22)\\ &\approx \mathcal{N}\Big(0, \sqrt{\beta_1^r(\alpha_1 x_r + \alpha_2) + \beta_2^r}\Big) \sim \epsilon_r(\alpha(x_r)). && (3.23) \end{aligned}$$

For details see Appendix A. We thus recover α from x_n and x_r by fitting the regression model

$$x_n \approx \operatorname{clip}\big(\alpha(x_r) + \epsilon_{r,n}(\alpha(x_r))\big), \tag{3.24}$$

where the parameters of the noise level function $\sigma_{r,n}$ of the compound noise $\epsilon_{r,n}$ are given by adding up the parameters $\boldsymbol{\beta}^r$ and $\boldsymbol{\beta}^n$ due to ϵ_r and ϵ_n being independent:

$$\begin{aligned} \epsilon_{r,n}(x_r) &\sim \mathcal{N}\big(0, \sigma_{r,n}(x_r)\big) && (3.25)\\ \text{with}\quad \sigma_{r,n}^2(x_r) &= (\beta_1^r + \beta_1^n)x_r + (\beta_2^r + \beta_2^n). && (3.26) \end{aligned}$$

Since the model defined in Eqs. (3.24) – (3.26) accounts for both clipped observations as well as the heteroscedasticity of the noise, we call it *heteroscedastic Tobit regression*.

It generalizes basic Tobit regression (Tobin, 1958), which only models clipped observations with homoscedastic noise. We can estimate the linear scaling parameters α_1, α_2 and the added noise variance parameters $\boldsymbol{\beta}^r + \boldsymbol{\beta}^n$ by maximizing the log-likelihood (see Appendix A). In Section 3.6 we demonstrate that faithful modeling of the image formation process with heteroscedastic Tobit regression is crucial for obtaining accurate estimates of α_1, α_2. Having recovered $\boldsymbol{\alpha}$ from the

unmasked pixels, we use it to linearly transform the intensities of the low-ISO image to get an intermediate post-processed image

$$y'_p = \alpha(x_r) = \alpha_1 x_r + \alpha_2. \tag{3.27}$$

Figure 3.6c shows the difference image after the linear correction. The intensity-dependent bias is removed.

Since the noise parameters $\boldsymbol{\beta}^n, \boldsymbol{\beta}^r$ mainly depend on the ISO value and characteristics of the sensor (Eq. 3.8c, European Machine Vision Association (2012)), we record them in a controlled laboratory setting using our regression model, see Section 3.6.3. Hence, for post-processing our real dataset, we fix $\boldsymbol{\beta}^r$ as well as $\boldsymbol{\beta}^n$ and only recover $\boldsymbol{\alpha}$. In Section 3.6.3 we demonstrate the accuracy of our noise estimates by showing that they are in high agreement to those obtained from spatial averaging (Foi et al., 2008).

SPATIAL MISALIGNMENT. We treat minuscule shifts of the camera as a global 2D translation that we wish to undo. While we have experimented with modern DFT-based subpixel alignment (Guizar-Sicairos et al., 2008), we found that the classical Lucas-Kanade approach (Lucas and Kanade, 1981) works better. Despite its simplicity, it recovers the translation very well even under strong noise, see Section 3.6. Having estimated the translation parameters from the unmasked pixels, we shift y'_p using bilinear interpolation to obtain the next intermediate image y''_p. Note that interpolation results in some smoothing. This is not critical when translating y'_p, since it contains few high frequencies. We avoid interpolating x_n as it contains many high frequencies due to the noise.

LOW-FREQUENCY RESIDUAL CORRECTION. As we can see in Fig. 3.6c, there remains a low-frequency pattern on the debiased residual image $R(y''_p)$. We account that to small changes in the ambient lighting. Also, when taking pictures under artificial illumination the rolling shutter effect will cause flickering of the light sources to appear as low-frequency banding artifacts. Thanks to the noise on the unmasked pixels being zero-mean in the debiased domain we can estimate the low-frequency pattern LF by low-pass filtering of $R(y''_p)$:

$$\mathrm{LF} = \mathrm{smooth}\left(R(y''_p)\right) = \mathrm{smooth}\left(\mathcal{A}(y''_p) - x_n\right). \tag{3.28}$$

The final post-processed image y_p is obtained by subtracting the low-frequency pattern and inverting the debiasing step as

$$y_p = \mathcal{A}^{-1}\left(\mathcal{A}(y''_p) - \mathrm{LF}\right). \tag{3.29}$$

We use a guided filter (He et al., 2013) with a large 40 pixel support for smoothing, which we found to remove structured residuals better

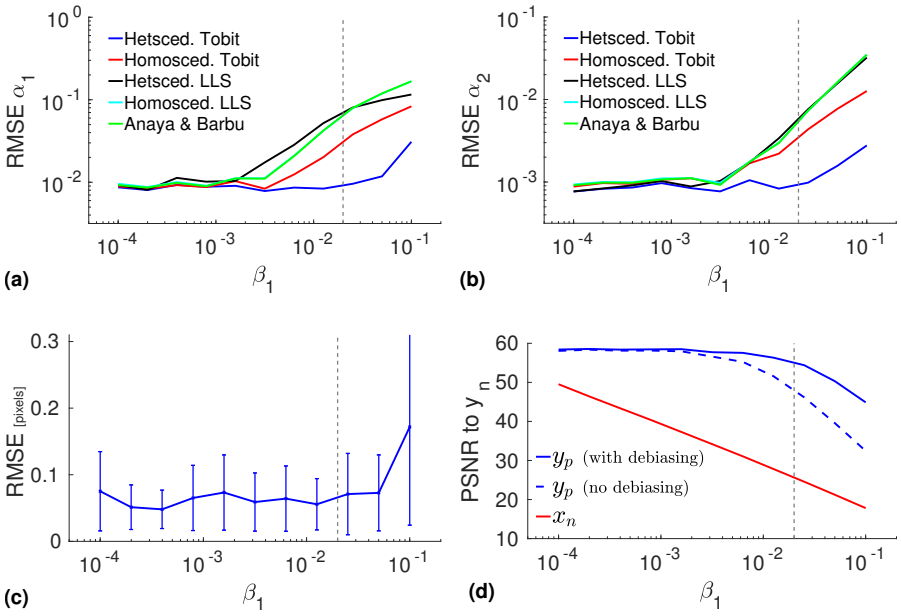


Figure 3.7. root mean squared error (RMSE) of recovering the slope (a) and offset (b) of simulated linear intensity scaling, and of recovering translation (c). PSNR to y_n for the post-processed reference image y_p and the noisy image x_n (d). The x-axes show the strength of the intensity-dependent noise, with real values in our benchmark lying left of the gray dashed line.

than a Gaussian filter in case α_1 is not estimated perfectly. Figure 3.6d shows the final debiased residual image $R(y_p)$ after the low-frequency correction. Now we can see a mostly zero-mean noise image as we expected, *cf.* Eq. (3.14). While the filtering adds some structured residuals tightly localized along strong edges, the magnitude of the effect is small compared to the noise strength. Also, we see that the variance of the noise increases with the image intensity, as expected for heteroscedastic noise.

3.6 EXPERIMENTAL VALIDATION

We now analyze and validate our approach on simulated data and demonstrate generalization to real image pairs.

3.6.1 *Post-processing is effective*

We first evaluate how accurately our post-processing can recover the transformation between the latent images y_n and y_r. Therefore, we simulate the image formation process of the reference and noisy image (Fig. 3.4). Specifically, we use captured low-ISO images as latent images y_n and generate the other latent image y_r by sampling a random transformation consisting of a spatial translation, linear intensity changes, and an additive low-frequency pattern. From the

latent images we generate the observations x_n and x_r by adding noise and clipping the image intensities. For realistic sampling of the transformations, we leverage statistics estimated on the captured dataset. Specifically, we sample random horizontal and vertical translations from $\mathcal{N}(0, 0.5)$. The slope and offset of the linear transformation are sampled from $\mathcal{N}(1, 0.05)$ and $\mathcal{N}(0, 0.0025)$, respectively. We generate the low-frequency pattern by sampling random Fourier coefficients weighted with a peaky Gaussian. We normalize the pattern in the spatial domain to have zero mean and a mean magnitude of 0.001. We finally simulate x_n and x_r by applying clipped Poisson-Gaussian noise to y_n and y_r, respectively, *i. e.*

$$x_i \sim \text{clip}\left(\beta_1^i \mathcal{P}(y_i/\beta_1^i) + \mathcal{N}\left(0, \sqrt{\beta_2^i}\right)\right), i \in \{n, r\}. \tag{3.30}$$

To validate the estimation accuracy for a wide range of scenarios, we evaluate 11 different parameter settings for the noise, with β_1^n ranging from 10^{-4} to 10^{-1} and β_2^n ranging from $5 \cdot 10^{-8}$ to 10^{-2}. This covers the range of noise parameters of the consumer cameras used for our dataset. For the reference image we use the noise level function of the Sony A7R at base ISO, *i. e.* $\beta_1^r \approx 2 \cdot 10^{-5}$, $\beta_2^r \approx 10^{-8}$. For each setting of noise parameters we run 100 trials in total.

We now study how well the proposed post-processing can undo the simulated transformations. First, we look at intensity scaling. Figures 3.7a and 3.7b show the RMSE of the estimated slope α_1 and offset α_2 of the linear intensity transformation. We compare our proposed Tobit regression model to several baselines: First, Tobit regression with homoscedastic noise (Tobin, 1958), *i. e.* the noise strength is independent of image intensities. Next, homoscedastic and heteroscedastic linear least squares where the observations are assumed to be unclipped. Finally, we compare to the regression model of (Anaya and Barbu, 2018), which models clipped observations while ignoring the intensity-dependence of the noise. We make two main observations: First, for low noise levels all methods perform equally well since the difficulty of the estimation problem is dominated by the other transformations, *i. e.* translation and low-frequency bias. Second, for medium to high noise levels our Tobit regression significantly outperforms all baselines including (Anaya and Barbu, 2018). This shows the importance of modeling the clipped, heteroscedastic observation process faithfully.

Next we turn to alignment. Figure 3.7c shows the RMSE in pixels for recovering the simulated translation. As can be seen, the estimation error is robust to increasing levels of noise as it remains roughly constant over most of the range of noise settings. The error increases only for severe noise.

Finally, we evaluate the removal of low-frequency bias. Figure 3.7d shows the PSNR between the post-processed image y_p and the latent image y_n. We compare our post-processing to a baseline (dashed) that

IS	A	LFC	Corr$(R(y_p), \mathcal{A}(y_p))$		Auto-Corr$(R(y_p))$		Var$(R(y_p))$ [$*10^{-3}$]	
			synth	real	synth	real	synth	real
			0.2144	0.1874	0.1407	0.1270	0.1921	0.1815
✓			0.0305	0.0318	0.0923	0.0843	0.1752	0.1690
	✓		0.2093	0.1892	0.0958	0.1024	0.1482	0.1583
✓	✓		0.0418	0.0474	0.0478	0.0560	0.1387	0.1473
		✓	0.0170	0.0175	0.0615	0.0581	0.1659	0.1626
✓		✓	0.0078	0.0067	0.0610	0.0559	0.1656	0.1621
	✓	✓	0.0118	0.0140	0.0066	0.0198	0.1313	0.1389
✓	✓	✓	0.0029	0.0051	0.0067	0.0173	0.1314	0.1385

Table 3.2. Statistics of the residual noise image for different combinations of post-processing steps on both synthetic and real data. For reference: $\text{Var}(R(y_n)) = 0.1222 \cdot 10^{-3}$, respectively $\text{Var}(R(y_n)) = 0.1356 \cdot 10^{-3}$ when sampling noise for x_n from $\epsilon_{r,n}$ instead of ϵ_n.

IS	A	LFC	PSNR(y_p, y_n), [dB]
			43.14
✓			46.45
	✓		46.37
✓	✓		51.18
		✓	47.31
✓		✓	47.56
	✓	✓	53.13
✓	✓	✓	53.71

Table 3.3. PSNR values when evaluating the post-processed reference image against the optimal ground truth image. The experiment is conducted on simulated data.

omits the debiasing step of Eqs. (3.28) to (3.29). Especially for high noise levels, the PSNR of the baseline is significantly lower, emphasizing that filtering in the debiased domain is important. We note that the PSNR of y_p reduces with higher noise levels, since the filtering step is not perfect and thus leaks low frequencies of the noise into the post-processed image. This is not critical, however, since the gap of the PSNR of the noisy image x_n to that of the latent image is still large enough to accurately measure state-of-the-art denoising performance.

3.6.2 *Quality of ground truth*

We now demonstrate that our post-processing pipeline provides accurate denoising ground truth on our real-world dataset by considering statistics of the debiased residual images. We have already seen that the *ground truth residual* $R(y_n)$ has mean zero given y_n. It follows

that $R(y_n)$ and $\mathcal{A}(y_n)$ are linearly uncorrelated (for a proof, see Appendix A). Furthermore, when assuming pixel-wise independent noise, $R(y_n)$ has zero auto-correlation. We thus expect the post-processing residual $R(y_p)$ to have small linear correlation to $\mathcal{A}(y_p)$ as well as small auto-correlation. Moreover, we expect $R(y_p)$ to have a slightly higher variance than $R(y_n)$, since $R(y_p)$ also includes the small amount of noise that affects x_r.

We evaluate the three statistics of $R(y_p)$ on our real-world dataset as well as on simulated data. To make the simulation as realistic as possible, for each image we use the parameters for translation and intensity scaling that were obtained by running post-processing on the real data and use the corresponding noise level functions. Table 3.2 shows the mean absolute linear correlation coefficient $\text{Corr}(R(y_p), \mathcal{A}(y_p))$, the mean absolute auto-correlation $\text{Auto-Corr}(R(y_p))$, and the geometric mean of the variance $\text{Var}(R(y_p))$. We observe a significant linear correlation when not applying any post-processing to x_r. Our full post-processing pipeline almost completely removes the correlation as expected from theoretical considerations, highlighting its need for obtaining a database with realistic image noise. Note that just applying the high-pass filter on the residual image (5^{th} row) still leaves a significant linear correlation, and that the combination of all three post-processing steps improves upon using any two post-processing steps. The same holds for auto-correlation, where our post-processing successfully obtains a residual image with low auto-correlation, indicating that the pixels in the noisy residual image are not highly spatially correlated. It is important to note that when only intensity scaling is applied, the auto-correlation is $5\times$ as high on real data. Since this is the only form of post-processing in the RENOIR dataset (Anaya and Barbu, 2018), we can conclude that our approach leads to a much more realistic image noise dataset.

Turning to variance, we see that the variance of the post-processing residual $R(y_p)$ is significantly closer to that of the ground truth residual $R(y_n)$ when all steps are carried out. The remaining gap to the ground truth residual can be explained as follows: The post-processed residual is affected by noise in x_r *and* x_n, while the ground truth residual is affected only by the noise in x_n. We thus also computed the variance of the ground truth residual $R(y_n)$ for a second setting, where we sample the noise of x_n from the compound noise $\epsilon_{r,n}$ (Eq. 3.25) instead of ϵ_n. Then the difference in variance between post-processed and ground truth residual almost vanishes, and the relative variance error decreases by an order of magnitude compared to no post-processing. This demonstrates that our post-processing removes the global effects on the residual image while accurately preserving the noise characteristics.

Importantly, the three test statistics obtained from synthetic experiments differ only marginally from those evaluated on the real captured

images, showing that the modeled transformation process consisting of translations, intensity scaling, and an additive low-frequency pattern accurately describes the real transformation between y_n and y_r.

Finally, Table 3.2 also shows the PSNR between y_n and y_p on simulated data. We see that our full post-processing achieves the highest PSNR of 53.7 dB. This is significantly more than what state-of-the-art denoising algorithms can currently achieve (Section 3.7), leaving enough room for measuring future improvements in terms of PSNR.

3.6.3 *Recording of noise parameters*

We calibrate the noise parameters $\boldsymbol{\beta}^r$ and $\boldsymbol{\beta}^n$ on controlled test scenes of a color checker. To estimate $\boldsymbol{\beta}^r$, we first run Tobit regression on pairs of images, both taken at base ISO, which yields an estimate of $2\boldsymbol{\beta}^r$ (Eq. 3.26). We subsequently recover $\boldsymbol{\beta}^n$ for all other ISO values by estimating $\boldsymbol{\beta}^r + \boldsymbol{\beta}^n$ on a low/high-ISO image pair and afterwards subtracting $\boldsymbol{\beta}^r$. To assess the accuracy of our estimates we compare them to those obtained from the individual images using the spatial averaging method of (Foi et al., 2008), which is designed to work highly accurately on images with piecewise constant intensities. We assess the agreement of both methods with the normalized RMSE Φ proposed in (Mäkitalo and Foi, 2014). It measures the relative error of standard deviations, averaged over pixel intensities. Specifically, we use the symmetric extension

$$\tilde{\Phi}(\boldsymbol{\beta}, \hat{\boldsymbol{\beta}}) = \tfrac{1}{2}\big(\Phi(\boldsymbol{\beta}, \hat{\boldsymbol{\beta}}) + \Phi(\hat{\boldsymbol{\beta}}, \boldsymbol{\beta})\big). \tag{3.31}$$

The mean error between Tobit regression and (Foi et al., 2008) is 0.003, *i. e.* standard deviations from both methods disagree only marginally by 0.3% on average. We conclude that Tobit regression produces accurate noise estimates on real data. But unlike spatial averaging methods, it generalizes to arbitrary scenes without large homogeneous areas.

We now justify using calibrated noise parameters for post-processing by showing that the noise parameters mainly depend on ISO value and camera, but not on absolute exposure time. For fixed combinations of ISO and camera, we estimate $\boldsymbol{\beta}^r$ and $\boldsymbol{\beta}^n$ across a range of exposure times of the image pairs. The average error $\tilde{\Phi}$ between those noise estimates is only 0.5%, showing that they are stable *w. r. t.* overall exposure times.

3.7 BENCHMARK

The proposed DND benchmark for denoising algorithms consists of 50 scenes selected from our captured images. We chose images that look like typical photographs, but also included images with interesting structures that we believe to be challenging for the algorithms tested.

Applied on	RAW		RAW+VST		sRGB
Evaluated on	RAW	sRGB	RAW	sRGB	sRGB
WNNM	46.29	37.64	47.10	**37.97**	34.44
KSVD	45.53	36.69	46.86	37.72	**36.55**
EPLL	46.34	37.27	46.85	37.55	33.51
FoE	45.77	36.09	44.11	35.97	34.49
NCSR	42.86	30.97	47.06	37.85	33.81
BM3D	**46.63**	**37.86**	**47.14**	37.95	34.61
MLP	42.70	33.74	45.70	36.83	34.14
TNRD	44.98	35.69	45.69	36.22	29.92

Table 3.4. Mean PSNR (in dB) of the denoising methods tested on our DND benchmark. We apply denoising either on linear raw intensities, after a VST, or after conversion to the sRGB space. Likewise, we evaluate the result either in linear raw space or in sRGB space. The noisy images have a PSNR of 39.39 dB (linear raw) and 29.98 dB (sRGB).

A subset of the test images is shown in Fig. 3.2. Table 3.1 lists the number of scenes per camera included in the benchmark dataset.

For the task of (non-blind) denoising, we compare the performance of Weighted Nuclear Norm Minimization (WNNM) (Gu et al., 2014), K-SVD (Aharon et al., 2006), Expected Patch Log Likelihood (EPLL) (Zoran and Weiss, 2011), Field of Experts (FoE) (Roth and Black, 2009) with the filters of (Gao and Roth, 2012), Nonlocally Centralized Sparse Representations (NCSR) (Dong et al., 2013), and BM3D (Dabov et al., 2007). Moreover, we benchmark two discriminative, "deep" methods: A multilayer network (MLP) (Burger et al., 2012) and Trainable Nonlinear Reactive Diffusion (TNRD) (Chen et al., 2015c). For MLP, we use available trained models for Gaussian noise with $\sigma \in \{10, 25, 35\}$. TNRD is trained on 400 separate images (Chen et al., 2015c) using code from the authors' web page. We train 10 models with different Gaussian noise standard deviations, evenly distributed in log-space from 0.0001 to 0.1, thus covering a reasonable range of noise levels observed on our real-world dataset.

We apply all algorithms to the noisy images in three different spaces. First, we use the space of linear raw intensities. Since the tested methods are mostly geared toward Gaussian denoising, we apply a VST prior to denoising as a second setting. This has the effect of approximately Gaussianizing the noise distribution. After retrieving the denoising result, we convert it back to linear raw space by applying an inverse VST. Specifically, we use the generalized Anscombe transform (Starck et al., 1998) and the closed-form approximation to its exact unbiased inverse (Mäkitalo and Foi, 2013). We parametrize the transformation with the noise-level functions obtained from the color-checker data (Section 3.6.3). In a third setting, we use available

Applied on	RAW		RAW+VST		sRGB
Evaluated on	RAW	sRGB	RAW	sRGB	sRGB
WNNM	0.971	**0.933**	**0.974**	**0.935**	0.866
KSVD	0.968	0.919	0.972	0.931	**0.900**
EPLL	0.968	0.931	0.973	0.927	0.829
FoE	0.967	0.907	0.955	0.914	0.887
NCSR	0.853	0.713	0.969	0.924	0.834
BM3D	**0.972**	**0.933**	**0.974**	0.932	0.855
MLP	0.939	0.886	0.963	0.916	0.838
TNRD	0.963	0.894	0.961	0.892	0.708

Table 3.5. Mean SSIM (Wang et al., 2004) of the denoising methods tested on our benchmark dataset. We apply denoising either on linear raw intensities, after a VST, or after conversion to the sRGB space. Likewise, we evaluate the result either in linear raw space or in sRGB space. The noisy images have a SSIM of 0.863 (linear raw) and 0.710 (sRGB).

EXIF data to simulate the main steps of the camera processing pipeline (Karaimer and Brown, 2016) that converts linear raw intensities to sRGB intensities. After white-balancing, we demosaic the image by linear interpolation. Finally, we convert from the camera internal color space to sRGB and apply gamma correction.

Since many of the benchmarked algorithms are too slow to be applied to megapixel-sized images, we crop 20 bounding boxes of 512×512 pixels from each image in the dataset, yielding 1000 test crops in total. They overlap at most 10% and do not contain pixels that were annotated as changing between the two exposures. We provide the algorithms with an estimate of the global noise standard deviation $\bar{\sigma}$ by computing the standard deviation of the residual noise image $R(y_p)$ on each crop. As the different color channels usually look quite distinct, we denoise each channel separately. For TNRD and MLP we choose the model whose σ for training is closest to the ground truth $\bar{\sigma}$. For FoE and EPLL we use a heteroscedastic Gaussian data term when denoising raw pixel intensities and a homoscedastic Gaussian data term in the other cases. For evaluation, we compare the denoised result to the post-processed reference image y_p either in linear raw space or in the sRGB space.

Table 3.4 and Table 3.5, respectively, show the PSNR and SSIM values (Wang et al., 2004) values, averaged over all crops and color channels. Looking at the PSNR values we make several interesting observations. As we can see, BM3D is overall the best performing method followed by WNNM. The other methods perform worse. The general tendency also holds across noise levels. This is quite surprising as the by now classic BM3D approach was previously considered to have been outperformed by the other approaches; our realistic noise dataset shows

that this is not the case. The discriminative methods fall short, which suggests that they generalize poorly to noise distributions that were not used during training. The generative FoE model performs surprisingly competitive in linear raw space, but is the only baseline that performs worse after VST. This suggests that FoE benefits from the more realistic likelihood in linear raw space.

Furthermore, we see that denoising sRGB images yields significantly worse results than applying denoising algorithms in raw space, since the noise distribution in sRGB space is spatio-chromatically correlated (Park et al., 2009). Another observation is that the amount of noise in our realistic dataset is lower than what is often used in the scientific literature for evaluating denoising algorithms using synthetic noise. The mean PSNR of the noisy images in raw space is 39.38 dB, which would correspond to a mean noise standard deviation of $\sigma \approx 2.74$ for images with intensities in $[0, 255]$. For comparison, most denoising algorithms are evaluated with noise standard deviations of at least $\sigma = 10$, which we believe to be mostly a historical artefact. Apparently, it was never really questioned whether they are still appropriate. Looking at SSIM values, we can see that BM3D and WNNM show the best performance and their scores differ only marginally. Overall, we observe that SSIM scores are high across all methods.

Finally, Figures 3.8 and 3.9 show denoising results of the tested algorithms for one crop of two different images in our database. The results were obtained from denoising raw intensities after the VST. We display the denoised images in sRGB space after our camera processing pipeline, *cf.* Section 3.7. We can see that many methods oversmooth fine structures (*e. g.*, MLP and FoE), while TNRD undersmoothes and fails to remove a significant part of the noise. Moreover, we can see that denoising introduces visually apparent color artifacts for all methods and that the noise is clearly spatio-chromatically correlated in sRGB space.

3.8 USAGE OF BENCHMARK

Here we want to discuss some of the effects that our DND benchmark had on the image denoising community. Let us start by outlining the way researchers can interact with our data. We released a website for the denoising benchmark in June 2017, making available an online submission system where users can register and upload their denoising results. These get automatically evaluated on our compute infrastructure which computes PSNR and SSIM values for each submission. As of March 2019, there are ~1300 registered users which uploaded a total of ~1600 submissions to our website. The user can choose between three levels of visibility of the results. The default option is to set the submission as "private", it is only visible to the submitting user. It can also be set to "private-anonymous", *i. e.* the quantitative results

appear on the public benchmark website while further information on the submission, such as authors, venue, description remain hidden. Eventually a submission can be set to "public", meaning that all information are visible on the public benchmark page. We furthermore restricted the number of submissions to 10 per month.

The submission limit and the choice to not disclose the post-processed reference images were made in order to avoid overfitting on the test data. It is a well known pitfall in machine learning contests, that participants are likely to observe increased scores on the public leader board when they submit often[3]. However, the increased accuracy on the public leaderboard is probably due to better fitting the noise in the public data rather than due to an improved generalization capability of the predictive model. Hence, it is common practice in these competitions to evaluate final results on a further test set where labels are not disclosed to the participants and where results can not be queried beforehand. Since we aim to test for generalization rather than overfitting we also decided to not disclose labels of our benchmark images. As there is no competition attached to our benchmark and hence also no predefined end date, we restrict the number of monthly submissions to limit possibilities of overfitting to the public leaderboard. Note, that there are approaches based on differential privacy that allow to retain the statistical validity of a test set even when querying it multiple times (Dwork et al., 2015a; b). However, these methods augment the usual empirical risk with some form of stochasticity, leading to stochastic evaluation results on the test set – a circumstance that would arguably hinder the adoption of our benchmark in the image denoising community.

We will now review the progress that has been made on our benchmark by fellow researchers. Table 3.6 shows the recent results (as of March 2019) on our public leaderboard. Specifically, results are shown for the CNN based approache DnCNN (Zhang et al., 2017a), UPI (Brooks et al., 2019), FFDNet (Zhang et al., 2018), CBDNet (Guo et al., 2019), and our N3Net (*cf.* Chapter 5) as well as the model based approaches TWSC (Xu et al., 2018a) and MCWNNM (Xu et al., 2017a). The leaderboard entries NLH+, NLH, MCAR, and DSSNet are still anonymous. We first note, that there is no algorithm that explicitely uses a variance stabilizing transformation, although learning based approaches might compute such a transformation implicitely. Most submissions were made on the sRGB denoising task. We can observe that the accuracy of the top-performing methods significantly improved over our initally benchmarked methods. The PSNR of denoised raw images increased by $1.7dB$ and $2.4dB$ when evaluated in linear raw space and sRGB space, respectively. The PSNR of denoised sRGB images increased by $2.3dB$. Still, the best accuracy on sRGB images is reached by a denoising method that operates on the linear raw inten-

3 http://gregpark.io/blog/Kaggle-Psychopathy-Postmortem/

Applied on	RAW		sRGB
Evaluated on	RAW	sRGB	sRGB
UPI (Raw)	**48.89**	40.17	–
UPI (sRGB)	48.88	**40.35**	–
N3Net	47.56	38.32	–
DSSNet	47.33	37.86	–
DnCNN	47.37	38.08	–
NLH+	–	–	**38.81**
NLH	–	–	38.79
MCAR	–	–	38.40
CBDNet	–	–	38.06
DSSNet	–	–	38.04
TWSC	–	–	37.94
FFDNet	–	–	37.61
MCWNNM	–	–	37.38
BM3D	47.14	37.95	34.61
KSVD	46.86	37.72	36.49

Table 3.6. Mean PSNR (in dB) of current top-performing submissions tested on our DND benchmark. The methods in the upper part of the table directly denoise linear raw intensities, whereas methods in the middle part of the table denoise sRGB intensities. For reference, in the lower part of the table we reproduce results of the top-performing methods among those that we intially benchmarked, *cf.* Table 3.4. We evaluate the result either in linear raw space or in sRGB space. The noisy images have a PSNR of 39.39 dB (linear raw) and 29.98 dB (sRGB).

sities, which is in line with our earlier findings. In contrast to MLP and TNRD, which we initially benchmarked, the leading learning based methods try to faithfully model the image formation process for training data generation. Moreover, both UPI and N3Net train only a single model for all noise levels in our dataset, demonstrating that the capacity of current deep architectures is large enough to interpolate between a broad range of noise levels. Figures 3.10 and 3.11 show example denoising results. We can visually observe that the quality of denoising improved considerably over the intial benchmarked algorithms in that more image detail is retained and fewer color artefacts are introduced. The success of deep learning based methods on the DND benchmark is mirrored in recent studies, *e. g.* the SIDD benchmark (Abdelhamed et al., 2018) that was also employed in the NTIRE 2019 challenge on image denoising (Abdelhamed et al., 2019a), where large training sets are available. The availability of training data in the SIDD dataset also facilitates goying beyond the

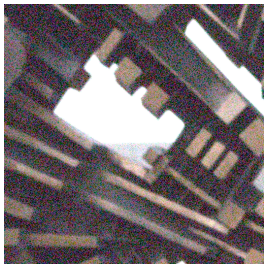
Poisson-Gaussian noise model that we studied for this work. Recent work fit deep generative models to the noise patterns of sRGB images, e.g. a GAN (Chen et al., 2018b) or normalizing flows (Abdelhamed et al., 2019b). Subsequently, they are able to generate synthetic noise patterns in order to augment the existing training set, thus improving the accuracy of deep neural network based denoisers.

3.9 CONCLUSION

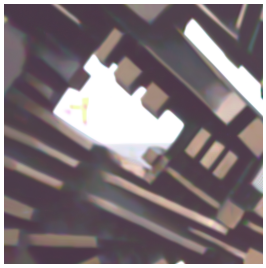
To benchmark denoising algorithms on real photographs, we introduced an acquisition procedure based on pairs of images of the same scene, captured with different analog gains and exposure time. While in theory the per-pixel mean intensity should stay constant, in practice we encountered residual errors. To derive ground-truth data, we proposed and evaluated a procedure for handling residual errors stemming from inaccurate gain and exposure time changes, relying on a heteroscedastic Tobit regression model. We also correct for lighting changes in a transformed space, as well as spatial misalignments. Our experiments showed the efficacy of this post-processing on simulated data, as well as its necessity on real photographs. We will make our novel ground-truth dataset of real photographs publicly available as a benchmark. We used it for evaluating various denoising algorithms and observed that BM3D continues to outperform recent denoising methods on real photographs, which is in contrast to findings on previously considered synthetic settings. More generally, our analysis revealed that the common scientific practice for evaluating denoising techniques has rather limited relevance for realistic settings.

(a) Full-sized reference

(b) Reference



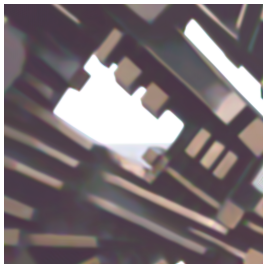
(c) Noisy (19.17 dB)



(d) WNNM (30.85 dB)

(e) KSVD (29.76 dB)

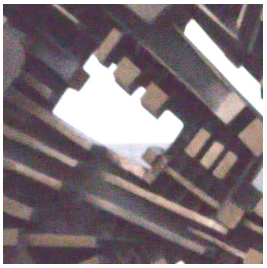
(f) EPLL (30.26 dB)



(g) NCSR (31.90 dB)

(h) BM3D (31.36 dB)

(i) MLP (31.38 dB)



(j) TNRD (26.51 dB)

(k) FoE (27.81 dB)

Figure 3.8. Example denoising result with PSNR values, displayed in sRGB space.

(a) Full-sized reference

(b) Reference

(c) Noisy (23.74 dB)

(d) KSVD (35.21 dB)

(e) WNNM (34.66 dB)

(f) EPLL (34.83 dB)

(g) NCSR (35.23 dB)

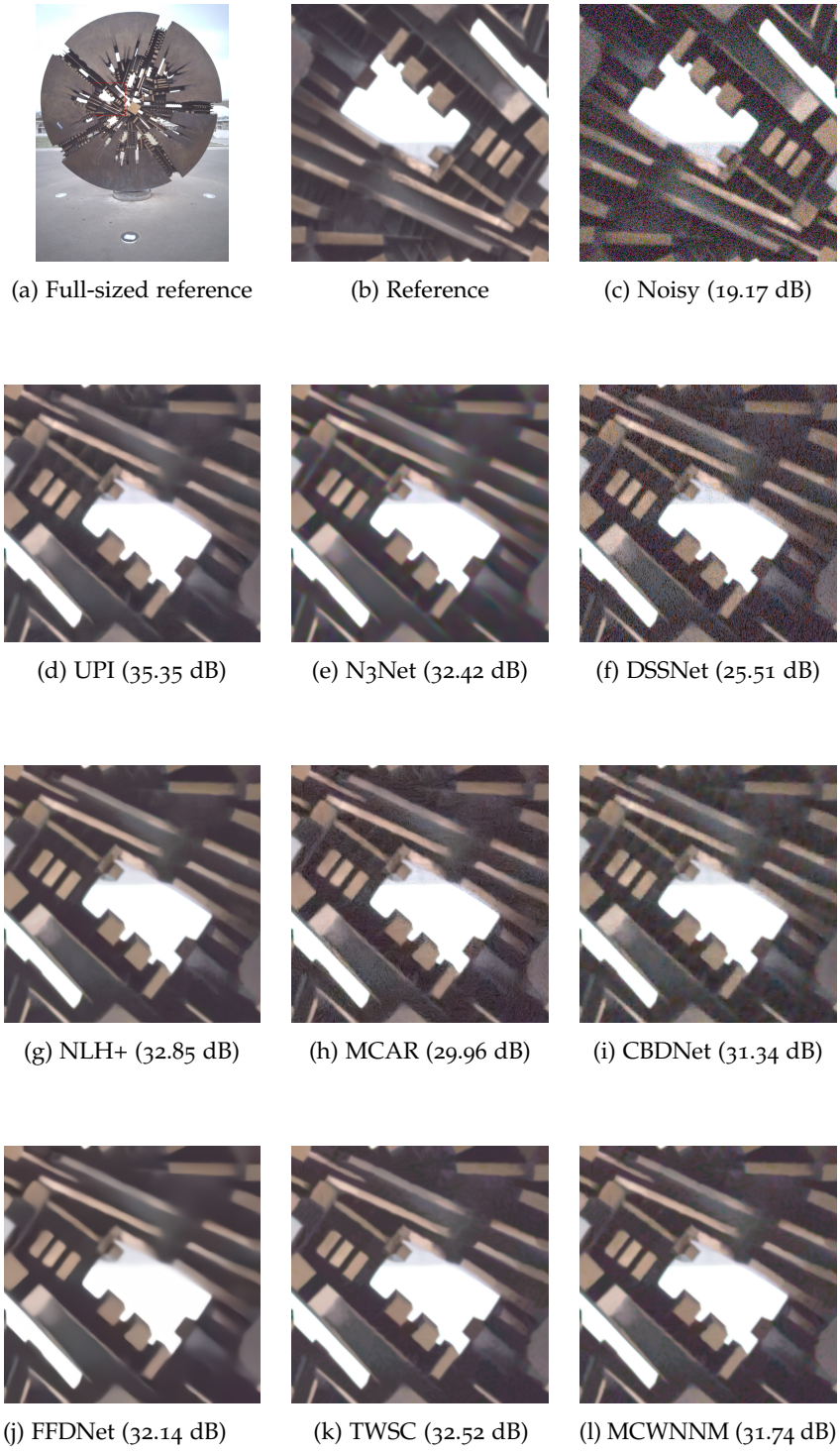
(h) BM3D (35.37 dB)

(i) MLP (35.43 dB)

(j) TNRD (31.94 dB)

(k) FoE (34.00 dB)

Figure 3.9. Example denoising result with PSNR values, displayed in sRGB space.



(a) Full-sized reference (b) Reference (c) Noisy (19.17 dB)

(d) UPI (35.35 dB) (e) N3Net (32.42 dB) (f) DSSNet (25.51 dB)

(g) NLH+ (32.85 dB) (h) MCAR (29.96 dB) (i) CBDNet (31.34 dB)

(j) FFDNet (32.14 dB) (k) TWSC (32.52 dB) (l) MCWNNM (31.74 dB)

Figure 3.10. Example denoising result with PSNR values of top-performing submissions, displayed in sRGB space. For UPI and DSSNet we show results of submission "UPI (sRGB)" and "DSSNet (sRGB)", respectively.

(a) Full-sized reference

(b) Reference

(c) Noisy (19.17 dB)

(d) UPI (37.73 dB)

(e) N3Net (35.35 dB)

(f) DSSNet (34.75 dB)

(g) NLH+ (35.04 dB)

(h) MCAR (36.08 dB)

(i) CBDNet (35.43 dB)

(j) FFDNet (34.47 dB)

(k) TWSC (34.53 dB)

(l) MCWNNM (34.43 dB)

Figure 3.11. Example denoising result with PSNR values of top-performing submissions, displayed in sRGB space. For UPI and DSSNet we show results of submission "UPI (sRGB)" and "DSSNet (sRGB)", respectively.

Part II

DENOISING CONVENTIONAL IMAGES

4

STOCHASTIC VARIATIONAL INFERENCE WITH GRADIENT LINEARIZATION

While the last chapter treated the question how to *measure* denoising accuracy, the next two chapters will aim at *improving* denoising accuracy. To this end, this chapter presents contributions for generative approaches to denoising while Chapter 5 contributes a new discriminative method. We now start by motivating our generative approach.

Recent progress in optimization of Monte Carlo objective functions led to new techniques for variational inference in very general model classes. These inference schemes are particularly appealing for practitioners as they do not require model specific derivations of update equations. Instead they only require gradients of the log-posterior which can be obtained through automatic differentiation engines. However, for certain random-field models, which are widespread tools for low-level computer vision problems, gradient-based optimization is known to struggle for even inferring the MAP estimate while a specialized technique, called gradient linearization, converges faster and often to better local minima. In this chapter we lift gradient linearization from MAP estimation to variational inference, resulting in a novel algorithm – stochastic variational inference with gradient linearization (SVIGL). It is easy to apply, requiring only a linearization of the log-posterior gradient, and it is on par to or even outperforms competing optimizers for SVI in terms of convergence speed, robustness and quality of the solution, as exemplified for the problems of optical flow estimation, Poisson-Gaussian denoising and 3D surface reconstruction. This chapter is based on (Plötz et al., 2018).

4.1 INTRODUCTION

Computer vision algorithms are quickly becoming more and more mature leading to complex systems being assembled from different algorithmic blocks. However, when aggregating different building blocks into system it becomes fruitful to assess the uncertainty associated with the predicted outputs of each component. To quantify uncertainty, probability distributions provide a rigorous mathematical framework, and techniques of VI and Monte Carlo sampling are often used to conduct the actual computations, see (Wainwright and Jordan, 2008) for an extensive introduction. However, Monte Carlo methods usually require numerous samples to achieve low variance estimates, thus resulting in a slow inference speed. On the other hand, VI is faster

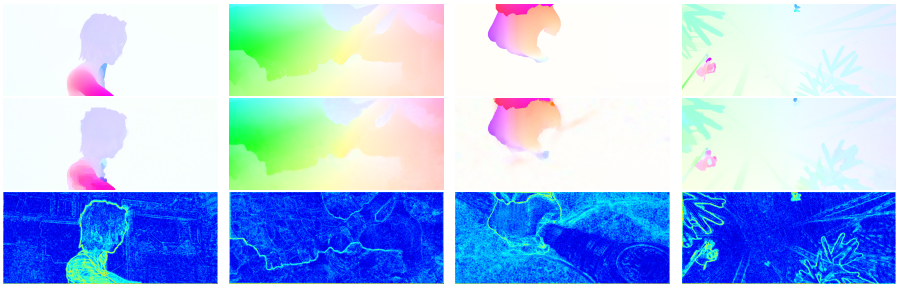


Figure 4.1. Variational optical flow estimation with SVIGL on an example image pair of Sintel final (Butler et al., 2012). Top: Ground truth flow map. Middle: Estimated posterior mean of flow map. Bottom: The estimated marginal uncertainty correlates well with errors of the flow prediction.

at inference time, yet the necessary update equations are usually very tedious to derive. Recent advances in stochastic, gradient-based optimization of the variational inference objective (Kingma and Welling, 2014; Ranganath et al., 2014; Rezende et al., 2014) have led to resurging interest in VI as they only require calculating the gradient of the logarithm of the posterior *w. r. t.* the unknown variables. This gradient can usually be derived automatically by automatic differentiation engines (Bischof et al., 1997). Thus, nowadays stochastic VI is part of probabilistic programming environments (Tran et al., 2017), allowing users to run inference for a large model-class without specifying model-specific update equations.

Not surprisingly, the generality of gradient-based SVI comes with the price of being sub-optimal for certain classes of probability distributions. Specifically, in this chapter we are considering highly non-convex posterior distributions, such as MRF models (Blake et al., 2011) that often arise in problems of low-level vision, *e. g.* optical flow (Brox et al., 2004; Revaud et al., 2015) or denoising (Roth and Black, 2011). For example, in optical flow the data term relates pixels or small patches from one frame to possible matching correspondences in the other frame, leading to a highly multi-model loss landscape. Further complicating things, the prior is usually chosen to be non-convex (Black and Anandan, 1991) in order to be robust to strong edges in the underlying flow field. Even for MAP inference, the multi-modal and non-convex posterior poses severe challenges to gradient-based optimization techniques. Hence, existing work (Brox et al., 2004; Revaud et al., 2015; Vogel and Oman, 1998) resort to another optimization technique, called optimization by gradient linearization (Nikolova and Chan, 2007). Here, instead of following the gradient at the current iterate, the gradient is linearized resulting in a linear system of equations. This corresponds to a quadratic approximation to the log posterior. The next iterate is obtained by solving for the minimum of this quadratic.

In this chapter we develop *stochastic variational inference with gradient linearization (SVIGL)* – a novel algorithm for SVI aiming at lifting the

benefits of optimization by gradient linearization from MAP inference to SVI. As a central result, we show that the gradient of a Monte Carlo approximation to the Kullback-Leibler (KL) divergence can be linearized conveniently as long as we have access to the gradient of the log posterior. As a result, this allows to use the machinery of gradient based optimization also for SVI. We provide further theoretic results demonstrating the soundness of our approach. Experiments on optical flow estimation and Poisson-Gaussian denoising show that SVIGL can compete or even outperform gradient-based SVI with regular stochastic gradient descent (SGD) and the strong ADAM optimizer (Kingma and Ba, 2015). Moreover, the choice of hyperparameters of SVIGL is found to be more robust than those of SVI with SGD or ADAM. An example flow field and associated uncertainty map obtained with SVIGL is shown in Fig. 4.1. We can see that errors in the flow field correlate well with high uncertainty, especially at motion discontinuities. SVIGL can be applied for more general problems which we exemplify by using it for variational point cloud denoising.

4.2 RELATED WORK

VARIATIONAL INFERENCE. To setup a VI problem it is necessary to specify a notion of distance between probability distributions and a class of approximating distributions q. The goal of VI is then to find the distribution q that is closest to the original, intractable distribution p in terms of the chosen distance. When considering the exclusive form of the KL divergence $\mathrm{KL}(q \,||\, p)$ the parametric form of approximating distributions is usually chosen such that the resulting updated equations are analytically tractable. Winn and Bishop (2005) present a general message passing scheme for VI in so-called conjugate-exponential models that are often used to describe topic models. One example is the well-known LDA model (Blei et al., 2003; Teh et al., 2008).

For applications in computer vision, MRF models are more common. Here, the application of VI has been restricted to certain model classes that allow for closed-form updates of the approximating distribution (*e. g.* Chantas et al., 2008; Levin et al., 2011; Likas and Galatsanos, 2004; Miskin and MacKay, 2000; Schelten and Roth, 2012). Miskin and MacKay (2000) used VI for Bayesian blind deconvolution. However, their model requires a fully factorized prior. Levin et al. (2011) define a mixture of Gaussian prior in the derivative space. However, they obtain a variational distribution over the clean image only while maintaining a point estimate of the kernel. Krähenbühl and Koltun (2011) show that mean field inference, *i. e.* VI *w. r. t.* the exclusive KL divergence with a fully factorized approximating distribution, can be done efficiently in MRF models with Gaussian edge potentials. For higher-order MRFs

Schelten and Roth (2012) apply VI to continuous MRFs with high-order GSM potentials.

All these works eventually employ an inference algorithm that is closely tied to the underlying posterior. Moreover, a slight change of the posterior model oftentimes necessitates a tedious re-derivation of closed-form update equations. In contrast, SVIGL allows for more practical Gaussian mean field VI as the only requirement on the true posterior model is a linearization of the gradient of the log posterior.

STOCHASTIC VARIATIONAL OPTIMIZATION. Recent works (*e. g.* Kingma and Welling, 2014; Rezende et al., 2014; Ruiz et al., 2016) have shown that we can use stochastic optimization for VI in general model classes as long as the approximating distribution q can be expressed as a deterministic, parameterized function of a random variable following some fixed base distribution. If q is reparameterizable in this way and the gradient of the log posterior can be computed as well, we can obtain Monte Carlo estimates of the gradient of the KL divergence (Kingma and Welling, 2014; Mnih and Rezende, 2016b). These works use SVI mainly for learning variational auto-encoders and other latent variable models. However, the inference technique is general and can be applied to other graphical models as well. Hence, gradient-based SVI is used as an inference backbone in recent frameworks for black-box VI (Im et al., 2017; Ranganath et al., 2014; Tran et al., 2017). Please note that we refer as SVI to the general problem of optimizing a stochastic estimator of a VI objective. The term is also used by Hoffman et al. (2013), where it relates to a technique for large-scale VI for posteriors from the conjugate-exponential family.

There are many different algorithms that can be employed for stochastic gradient-based optimization (Robbins and Monro, 1951). Besides the classical SGD modern choices comprise the strong and versatile ADAM optimizer (Kingma and Ba, 2015), RMSprop (Tieleman and Hinton, 2012), AdaGrad (Duchi et al., 2011), or L-BFGS-SGVI (Fan et al., 2015). These methods adaptively tune the step size of each dimension by looking at the statistics of recent gradient evaluations. This can be seen as calculating a special preconditioner for gradient descent. The gradient linearization update can also be interpreted as preconditioned gradient descent (Nikolova and Chan, 2007) where the preconditioner, in contrast to the previously mentioned methods, does only depend on the current iterate and not on the history of gradient evaluations.

APPLICATIONS OF UNCERTAINTIES. In computer vision, the approximate posterior q obtained from VI can be used to derive more robust predictions compared to simple MAP estimates (*e. g.* Krähenbühl and Koltun, 2011; Levin et al., 2011). However, we can also query the approximate posterior for an estimate of the uncertainty that is

associated with a prediction, *e. g.* by looking at the marginal entropy. These uncertainty estimates are useful in themselves as they allow to put more weight on the confident parts of a prediction, *e. g.* as done by the bilateral solver (Barron and Poole, 2016) ot to discard unreliable decisions altogether (Ochs et al., 2014; Wannenwetsch et al., 2017; Wedel et al., 2009). For video restoration, uncertainties have been found useful for fusing information across several frames (Chen and Tang, 2007).

4.3 PRELIMINARIES

In general, the goal of VI (Wainwright and Jordan, 2008) is to approximate an intractable distribution p by the closest distribution q that comes from a family of tractable distributions. Since we will validate our results on VI in CRF models, we will specifically look at posterior distributions $p(\mathbf{y} \mid \mathbf{x})$. However, our inference method applies to joint and marginal distributions as well.

Our approach applies to Gibbs distributions over continuous random variables, *i. e.* we can identify p by its energy function $E(\mathbf{y}, \mathbf{x})$:

$$p(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\Big\{ - E(\mathbf{y}, \mathbf{x}) \Big\}, \tag{4.1}$$

where we subsumed the temperature parameter into $E(\mathbf{y}, \mathbf{x})$ and where $Z(\mathbf{x})$ denotes the partition function given by

$$Z(\mathbf{x}) = \int \exp\Big\{ - E(\mathbf{y}, \mathbf{x}) \Big\} d\mathbf{y}. \tag{4.2}$$

We assume that $E(\mathbf{y}, \mathbf{x})$ is differentiable *w. r. t.* $\mathbf{y}$.

In this work, we parameterize the approximate posterior q by variational parameters $\boldsymbol{\theta}$ and look for optimal variational parameters $\hat{\boldsymbol{\theta}}$ such that the exclusive KL divergence is minimized, *i. e.*

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\min_{\boldsymbol{\theta}} \operatorname{KL}(q \,||\, p) && (4.3)\\
&= \arg\min_{\boldsymbol{\theta}} - \mathbb{E}_{q(\mathbf{y};\boldsymbol{\theta})}[\log p(\mathbf{y} \mid \mathbf{x})] + \mathbb{E}_{q(\mathbf{y};\boldsymbol{\theta})}[\log q(\mathbf{y};\boldsymbol{\theta})] && (4.4)\\
&= \arg\min_{\boldsymbol{\theta}} - \mathbb{E}_{q(\mathbf{y};\boldsymbol{\theta})}[\log p(\mathbf{y} \mid \mathbf{x})] - H(q), && (4.5)
\end{aligned}$$

where $H(q) = H\big(q(\mathbf{y};\boldsymbol{\theta})\big)$ denotes the entropy of q.

GRADIENT LINEARIZATION. We now turn to the technique of gradient linearization. In computer vision it is often used for finding

the MAP estimate $\hat{\mathbf{y}}$ (*cf.* Eq. 1.4) in continuous and differentiable energy models $E(\mathbf{y}, \mathbf{x})$, *e. g.* as defined in Eq. 4.1,

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} \log p(\mathbf{y} \mid \mathbf{x}) = \arg\min_{\mathbf{y}} E(\mathbf{y}, \mathbf{x}). \tag{4.6}$$

Note that we do not need to consider the log-partition function here as it is constant *w. r. t.* $\mathbf{y}$. Using standard gradient-based optimization to find $\hat{\mathbf{y}}$ may lead to slow convergence for certain MRF models in computer vision. Furthermore, the Hessian of the energy function may be dense or hard to compute, thus preventing the use of second-order optimization methods. Eventually, for energy minimization problems in computer vision, *e. g.* for optical flow (Brox et al., 2004; Revaud et al., 2015), denoising (Vogel and Oman, 1996), or deblurring (Vogel and Oman, 1998), an iterative inference technique called *gradient linearization* has received widespread attention. In each iteration of gradient linearization (GL), the gradient of the energy function E *w. r. t.* $\mathbf{y}$ gets linearized at the current estimate $\mathbf{y}^{(t)}$:

$$\nabla_{\mathbf{y}} E(\mathbf{y}, \mathbf{x}) \approx \bar{\nabla}_{\mathbf{y}} E\big(\mathbf{y}; \mathbf{y}^{(t)}\big) = \mathbf{A}_{\mathbf{y}}\big(\mathbf{y}^{(t)}\big)\mathbf{y} + \mathbf{b}_{\mathbf{y}}\big(\mathbf{y}^{(t)}\big). \tag{4.7}$$

To simplify notation, we omit $\mathbf{x}$ here and in the following. The linearized gradient can be related to a quadratic approximation of the energy function that has the same gradient as E at $\mathbf{y}^{(t)}$, *i. e.* the linearized gradient $\bar{\nabla}_{\mathbf{y}} E\big(\mathbf{y}; \mathbf{y}^{(t)}\big)$ is exact at $\mathbf{y} = \mathbf{y}^{(t)}$. The next iterate $\mathbf{y}^{(t+1)}$ is obtained by minimizing the quadratic approximation, meaning that we solve for the root of $\bar{\nabla}_{\mathbf{y}} E$. That amounts to solving the following linear equation system

$$\mathbf{y}^{(t+1)} = -\mathbf{A}_{\mathbf{y}}^{-1}\big(\mathbf{y}^{(t)}\big)\mathbf{b}_{\mathbf{y}}\big(\mathbf{y}^{(t)}\big). \tag{4.8}$$

Since this is an iterative procedure, we need to initialize the optimization by defining $\mathbf{y}^{(0)}$.

GL has connections to various other optimization schemes. Nikolova and Chan (2007) showed and equivalence of GL and the multiplicative form of half-quadratic minimization (Geman and Reynolds, 1992) for Gaussian likelihoods. Moreover, there is a close relationship to iteratively reweighted least squares through this equivalence (Idier, 2001). In Appendix B we show that GL can be interpreted as preconditioned gradient descent using $\mathbf{A}_{\mathbf{y}}^{-1}$ as preconditioner (Nikolova and Chan, 2007). Please note that in contrast to Newton's method GL does not require second-order derivatives. This is akin to quasi-Newton methods like L-BFGS (Byrd et al., 1995). In contrast to regular gradient descent every iteration of GL (Eq. 4.8) updates all variables jointly and interdependently. This enables faster convergence for highly multi-modal and non-convex objectives (*cf.* Fig. 4.2).

4.4 STOCHASTIC VARIATIONAL INFERENCE WITH GRADIENT LINEARIZATION (SVIGL)

We will now demonstrate how we apply GL to the Gaussian mean field VI problem assuming that we have access to the linearized energy gradient. We first apply the re-parameterization trick (Kingma and Welling, 2014; Rezende et al., 2014) to reformulate the exclusive KL divergence (Eq. 4.5) as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} - \mathbb{E}_{\mathbf{z}\sim\mathcal{G}}\Big[\log p\big(\mathbf{y}(\mathbf{z}) \mid \mathbf{x}\big)\Big] - H(q), \tag{4.9}$$

where $\mathbf{y}(\mathbf{z}) \equiv \mathbf{y}(\mathbf{z};\boldsymbol{\theta})$, and $\mathbf{z}$ follows a base distribution $\mathcal{G}$ that does not depend on $\boldsymbol{\theta}$. Using the linearization of the energy gradient given by $\mathbf{A}_\mathbf{y}$ and $\mathbf{b}_\mathbf{y}$ and a finite set of samples $\mathcal{Z} = \{\mathbf{z}_i\}$ we can now derive a Monte Carlo approximation of the gradient of the KL divergence in Eq. (4.9) with respect to the parameters $\boldsymbol{\theta}$:

$$\begin{aligned}
&\nabla_{\boldsymbol{\theta}} \mathrm{KL}\left(q \,||\, p\right) \\
&\overset{(4.9)}{=} - \mathbb{E}_{\mathbf{z}\sim\mathcal{G}}\Big[\nabla_\mathbf{y} \log p\big(\mathbf{y}(\mathbf{z}) \mid \mathbf{x}\big) \cdot \nabla_{\boldsymbol{\theta}}\, \mathbf{y}(\mathbf{z})\Big] - \nabla_{\boldsymbol{\theta}} H(q) && (4.10)\\
&\approx - \frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i\in\mathcal{Z}} \nabla_\mathbf{y} \log p\big(\mathbf{y}(\mathbf{z}_i) \mid \mathbf{x}\big) \cdot \nabla_{\boldsymbol{\theta}}\, \mathbf{y}(\mathbf{z}_i) - \nabla_{\boldsymbol{\theta}} H(q) && (4.11)\\
&\overset{(4.7)}{\approx} \frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i\in\mathcal{Z}} \Big(\mathbf{A}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big)\mathbf{y}(\mathbf{z}_i) + \mathbf{b}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big)\Big) \cdot \nabla_{\boldsymbol{\theta}}\, \mathbf{y}(\mathbf{z}_i) \\
&\qquad - \nabla_{\boldsymbol{\theta}} H(q) && (4.12)\\
&\equiv \bar{\nabla}_{\boldsymbol{\theta}} \mathrm{KL}\left(q \,||\, p\right). && (4.13)
\end{aligned}$$

GAUSSIAN MEAN FIELD INFERENCE. We now refine this approximation in the context of the common naive mean field framework (Chantas et al., 2008; Kingma and Welling, 2014; Levin et al., 2011). Here the variational distribution q factorizes along all elements of $\mathbf{y} = \big(x_l\big)_l$ for $l = 1, \ldots, L$ into independent one-dimensional Gaussian distributions. Denoting with $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \boldsymbol{\sigma}\}$ the variational parameters given by the mean parameters $\boldsymbol{\mu}$ and the standard deviation parameters $\boldsymbol{\sigma}$ we have:

$$q(\mathbf{y}) = \prod_{l=1}^{L} \mathcal{N}(x_l \mid \mu_l, \sigma_l^2). \tag{4.14}$$

We can now reparameterize $q(\mathbf{y})$ by chosing $\mathbf{z}$ to be drawn from standard normal distribution, *i.e.* $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$, and setting $\mathbf{y}(\mathbf{z}) = \mathbf{z} \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}$, where all operations are meant to be element-wise (Kingma and Welling, 2014).

In order to eventually apply GL we will now reformulate the gradient $\bar{\nabla}_{\theta} \mathrm{KL}\left(q \,||\, p\right)$ in a linearized way. To this end, let us look at the partial derivatives *w. r. t.* $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. Exploiting that the entropy of a Gaussian distribution is independent of its mean we can reformulate the gradient with respect to $\boldsymbol{\mu}$ as

$$
\begin{aligned}
&\bar{\nabla}_{\boldsymbol{\mu}} \mathrm{KL}\left(q \,||\, p\right) \\
=&\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \Big(\mathbf{A}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big)\mathbf{y}(\mathbf{z}_i) + \mathbf{b}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big) \Big) \cdot \nabla_{\boldsymbol{\mu}} \mathbf{y}(\mathbf{z}_i) \\
&- \nabla_{\boldsymbol{\mu}} H(q) && (4.15) \\
=&\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{A}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big)\big(\mathbf{z}_i \cdot \boldsymbol{\sigma} + \boldsymbol{\mu}\big) + \mathbf{b}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big) && (4.16) \\
=&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{A}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big)\right]\boldsymbol{\mu} \\
+&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{A}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big)\mathbf{D}(\mathbf{z}_i)\right]\boldsymbol{\sigma} \\
+&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{b}_{\mathbf{y}}\big(\mathbf{y}(\mathbf{z}_i)\big)\right] && (4.17) \\
\equiv&\mathbf{A}_{\boldsymbol{\mu},\boldsymbol{\mu}}(\boldsymbol{\theta})\ \boldsymbol{\mu} + \mathbf{A}_{\boldsymbol{\mu},\boldsymbol{\sigma}}(\boldsymbol{\theta})\ \boldsymbol{\sigma} + \mathbf{b}_{\boldsymbol{\mu}}(\boldsymbol{\theta}). && (4.18)
\end{aligned}
$$

Here $\mathbf{D}(\mathbf{z}_i)$ denotes a diagonal matrix that consists of the elements of $\mathbf{z}_i$.

Let us now turn to the gradient *w. r. t.* $\boldsymbol{\sigma}$. We now need to consider the derivative of the Gaussian entropy, *i. e.*

$$
\nabla_{\boldsymbol{\sigma}} H(q) = \nabla_{\boldsymbol{\sigma}} \log \boldsymbol{\sigma} + \text{const}, \tag{4.19}
$$

for which multiple different linearizations are possible. We use the element-wise second-order Taylor expansion of the logarithm around the current estimate: $\boldsymbol{\sigma}^{(t)}$:

$$
\begin{aligned}
\log \boldsymbol{\sigma} \approx& \log \boldsymbol{\sigma}^{(t)} + \frac{1}{\boldsymbol{\sigma}^{(t)}}\big(\boldsymbol{\sigma} - \boldsymbol{\sigma}^{(t)}\big) - \frac{1}{\big(\boldsymbol{\sigma}^{(t)}\big)^2}\big(\boldsymbol{\sigma} - \boldsymbol{\sigma}^{(t)}\big)^2 && (4.20) \\
=& \frac{1}{\boldsymbol{\sigma}^{(t)}}\boldsymbol{\sigma} - \frac{1}{\big(\boldsymbol{\sigma}^{(t)}\big)^2}\big(\boldsymbol{\sigma} - \boldsymbol{\sigma}^{(t)}\big)^2 + \text{const}. && (4.21)
\end{aligned}
$$

Now we are ready to obtain a linearized gradient of the Monte Carlo approximation to the KL divergence *w. r. t.* $\boldsymbol{\sigma}$:

$$
\begin{aligned}
&\bar{\nabla}_\sigma \mathrm{KL}\left(q \,||\, p\right) \\
=&\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \Big(\mathbf{A}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big)\mathbf{y}(\mathbf{z}_i) + \mathbf{b}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big) \Big) \cdot \nabla_\sigma \mathbf{y}(\mathbf{z}_i) \\
&- \nabla_\sigma H(q) && (4.22) \\
\approx&\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{D}(\mathbf{z}_i) \Big(\mathbf{A}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big)\left(\mathbf{z}_i \cdot \sigma + \mu\right) + \mathbf{b}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big) \Big) \\
&- \frac{3}{\sigma^{(t)}} + \frac{2}{\left(\sigma^{(t)}\right)^2}\sigma && (4.23) \\
=&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{D}(\mathbf{z}_i)\mathbf{A}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big) \right] \mu \\
+&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{D}(\mathbf{z}_i)\mathbf{A}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big)\mathbf{D}(\mathbf{z}_i) + \frac{2}{\left(\sigma^{(t)}\right)^2} \right] \sigma \\
+&\left[\frac{1}{|\mathcal{Z}|} \sum_{\mathbf{z}_i \in \mathcal{Z}} \mathbf{z}_i \mathbf{b}_\mathbf{y}\big(\mathbf{y}(\mathbf{z}_i)\big) - \frac{3}{\sigma^{(t)}} \right] && (4.24) \\
\equiv&\mathbf{A}_{\sigma,\mu}(\boldsymbol{\theta})\ \mu + \mathbf{A}_{\sigma,\sigma}(\boldsymbol{\theta})\ \sigma + \mathbf{b}_\sigma(\boldsymbol{\theta}). && (4.25)
\end{aligned}
$$

Using the results of Eqs. (4.18) and (4.25), we can derive the linearized gradient of the KL divergence in Eq. (4.5) with respect to μ and σ as

$$
\bar{\nabla}_{\boldsymbol{\theta}} \mathrm{KL}\left(q \,||\, p\right) = \mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta})\boldsymbol{\theta} + \mathbf{b}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) \tag{4.26}
$$

with

$$
\mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{A}_{\mu,\mu}(\boldsymbol{\theta}) & \mathbf{A}_{\mu,\sigma}(\boldsymbol{\theta}) \\ \mathbf{A}_{\sigma,\mu}(\boldsymbol{\theta}) & \mathbf{A}_{\sigma,\sigma}(\boldsymbol{\theta}) \end{bmatrix}, \ \mathbf{b}_{\boldsymbol{\theta}}(\boldsymbol{\theta}) = \begin{bmatrix} \mathbf{b}_\mu(\boldsymbol{\theta}) \\ \mathbf{b}_\sigma(\boldsymbol{\theta}) \end{bmatrix}. \tag{4.27}
$$

We can now use the GL framwork to optimize the KL divergence. In each iteration we obtain the next iterate $\boldsymbol{\theta}^{(t+1)}$ through solving the linear system of equations

$$
\boldsymbol{\theta}^{(t+1)} = -\mathbf{A}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(t)}\right)^{-1} \mathbf{b}_{\boldsymbol{\theta}}\left(\boldsymbol{\theta}^{(t)}\right) \tag{4.28}
$$

Note that the only interaction with the underlying energy E is through its linearized gradient. Apart from that, we treat E as a black box. Our approach is summarized in Algorithm 1.

DISCUSSION. As discussed earlier, solving for the root of the linearized gradient can be related to finding the optimum of a quadratic that has the same first-order derivate as the stochastic approximation of the KL divergence (Eq. 4.11) at the point $\boldsymbol{\theta}^{(t)}$. We now show that the

Algorithm 1 Gaussian mean field inference with SVIGL

Require: $\boldsymbol{\theta}^{(0)}$: Initial variational parameters
  $\mathbf{A}_\mathbf{y}$, $\mathbf{b}_\mathbf{y}$: Gradient linearization of the model energy
  for $t = 0, \dots, T-1$ **do**
    Generate samples $\mathbf{z}_i$
    $\mathbf{y}_i \leftarrow \boldsymbol{\sigma} \cdot \mathbf{z}_i + \boldsymbol{\mu}$
    Compute $\mathbf{A}_\mathbf{y}(\mathbf{y}_i)$ and $\mathbf{b}_\mathbf{y}(\mathbf{y}_i)$
    Compute $\mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$ and $\mathbf{b}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$ as in Eq. (4.27)
    $\boldsymbol{\theta}^{(t+1)} \leftarrow -\mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})^{-1}\mathbf{b}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$
  end for
  return $\boldsymbol{\theta}^{(T)}$

extremum of the quadratic is a actually a minimum by showing that the Hessian of the quadratic approximation is positive semi-definite.

Proposition 1. *$\mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$ is positive semi-definite,* i. e. *$\boldsymbol{\theta}^T\mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})\boldsymbol{\theta} \geq 0, \forall \boldsymbol{\theta}, \boldsymbol{\theta}^{(t)} \in \mathbb{R}^{2L}$, if the matrix $\mathbf{A}_\mathbf{y}(\mathbf{y}(\mathbf{z}))$ of the energy GL is positive semi-definite for all $\mathbf{y}(\mathbf{z})$.*

Proof. We first proof the proposition when drawing just a single sample $\mathbf{z}$. To simplify notation let $\mathbf{A}_\mathbf{y} \equiv \mathbf{A}_\mathbf{y}(\mathbf{y}(\mathbf{z}))$ and $\mathbf{A}_{\boldsymbol{\theta}} \equiv \mathbf{A}_{\boldsymbol{\theta}}(\boldsymbol{\theta}^{(t)})$. Now, for $\boldsymbol{\theta} = [\boldsymbol{\mu}, \boldsymbol{\sigma}]^T$ we have that

$$
\begin{aligned}
&\boldsymbol{\theta}^T\mathbf{A}_{\boldsymbol{\theta}}\,\boldsymbol{\theta} \\
&= \boldsymbol{\mu}^T\mathbf{A}_{\mu,\mu}\,\boldsymbol{\mu} + \boldsymbol{\sigma}^T\mathbf{A}_{\sigma,\mu}\,\boldsymbol{\mu} + \boldsymbol{\mu}^T\mathbf{A}_{\mu,\sigma}\,\boldsymbol{\sigma} + \boldsymbol{\sigma}^T\mathbf{A}_{\sigma,\sigma}\,\boldsymbol{\sigma} && (4.29)\\
&= \boldsymbol{\mu}^T\mathbf{A}_\mathbf{y}\boldsymbol{\mu} + \boldsymbol{\sigma}^T\mathbf{D}(\mathbf{z})^T\mathbf{A}_\mathbf{y}\boldsymbol{\mu} + \boldsymbol{\mu}^T\mathbf{A}_\mathbf{y}\mathbf{D}(\mathbf{z})\boldsymbol{\sigma} && (4.30)\\
&\quad + \boldsymbol{\sigma}^T\mathbf{D}(\mathbf{z})^T\left(\mathbf{A}_\mathbf{y} + \mathbf{D}\left(2/\left(\sigma^{(i)}\right)^2\right)\right)\mathbf{D}(\mathbf{z})\boldsymbol{\sigma} \\
&= \left(\boldsymbol{\mu} + \mathbf{D}(\mathbf{z})\boldsymbol{\sigma}\right)^T\mathbf{A}_\mathbf{y}\left(\boldsymbol{\mu} + \mathbf{D}(\mathbf{z})\boldsymbol{\sigma}\right) && (4.31)\\
&\quad + \left(\mathbf{D}(\mathbf{z})\boldsymbol{\sigma}\right)^T\mathbf{D}\left(2/\left(\sigma^{(i)}\right)^2\right)\left(\mathbf{D}(\mathbf{z})\boldsymbol{\sigma}\right) \\
&\geq 0. && (4.32)
\end{aligned}
$$

Here we plugged in the definition of the constituent matrices of $\mathbf{A}_{\boldsymbol{\theta}}$ (Eqs. 4.18 and 4.25). In the last step, we invoked the assumption that $\mathbf{A}_\mathbf{y}$ is positive semi-definite. For the case of drawing multiple samples $\mathbf{z}_i$ we can expand each of the four terms in Eq. (4.29) into a sum and use the fact that positive semi-definite matrices are closed under summation. □

The above proposition hinges on the assumption that the matrix $\mathbf{A}_\mathbf{y}$ is positive semi-definite. We now show that we can always obtain a positive semi-definite $\mathbf{A}_\mathbf{y}$ if the energy function fulfills two mild sufficient conditions.

Proposition 2. *An energy function can be linearized with a positive semi-definite matrix* $\mathbf{A}_\mathbf{y}$ *if it is composed of a sum of energy terms* $\rho_i(\mathbf{w}_i)$ *that fulfill the following conditions:*

1. *Each penalty function* $\rho_i(\cdot)$ *is symmetric and* $\rho_i'(\mathbf{w}_i) \geq 0$ *for all* $\mathbf{w}_i \geq 0$. $(\star)$
2. *Each penalty function* $\rho_i(\cdot)$ *is applied element-wise on* $\mathbf{w}_i$*, which is of the form* $\mathbf{w}_i = \mathbf{K}_i\mathbf{y} + \mathbf{g}_i(\mathbf{x})$*, with filter matrix* $\mathbf{K}_i$ *and function* $\mathbf{g}_i$ *not depending on* $\mathbf{y}$. $(\star\star)$

Proof. See Appendix B. □

Many MRF and CRF potentials meet the above two conditions (Blake et al., 2011), including GSM potentials in the FoE prior (Roth and Black, 2011) for optical flow or denoising. Also the data term of our flow energy (Section 4.5.1) is included in this class of potentials. It is also possible to obtain a positive semi-definite $\mathbf{A}_\mathbf{y}$ for more complex potentials such as the non-symmetric data term of the Poisson-Gaussian denoising energy, *cf.* Section 4.5.2. For other more flexible potential functions, such as radial basis function potentials (Schmidt and Roth, 2014), or periodic potentials, such as the cosine potential of the von-Mises distribution (*cf.* Chapter 6), the above conditions do not hold in general. Hence, $\mathbf{A}_\mathbf{y}$ might not be positive semi-definite for certain $\mathbf{y}$ in these cases.

IMPLEMENTATION DETAILS. Large-scale problems may involve millions of variables. Hence, solving the linear system of equations of Eq. (4.28) exactly might be too costly. Therefore, we use an approximate solver by applying 100 iterations of successive over-relaxation (Young, 1971) with a relaxation factor of 1.95, initializing with the current iterate $\boldsymbol{\theta}^{(t)}$. We empirically found a conjugate gradient optimizer to converge too slowly, probably due to requiring an effective preconditioner. The update of Eq. (4.28) does not guarantee that σ stays positive. Hence, we force this by replacing each new estimate $\sigma^{(t+1)}$ with its absolute value. However, we observed that the entropy term is usually sufficient to keep σ positive. Note, that we deliberately do not optimize for $\log\sigma$ since then the gradient of the KL divergence cannot conveniently be written as a linear function of $\log\sigma$.

4.5 EXPERIMENTS

We now show the versatility and efficiency of SVIGL by obtaining variational approximations from well-known energy functions for diverse low-level vision problems[1]. Besides delivering accurate mean estimates, their errors correlate well with the uncertainty as measured by the marginal entropies of the approximation. We specifically

1 Code is available at: https://github.com/tobiasploetz/SVIGL

consider two tasks for quantitative evaluation: Optical flow estimation and Poisson-Gaussian denoising. As common baselines we chose gradient-based stochastic optimization of the KL divergence with either SGD or the strong Adam optimizer (Kingma and Ba, 2015), which is widely used, *e. g.* as the default optimizer of the popular Edward library (Tran et al., 2017). We quantitatively evaluate the quality of the variational approximation, by computing a Monte Carlo estimate of the KL divergence $\mathrm{KL}\left(q \,||\, p\right)$ up to the unknown but constant log partition function $\log Z(\mathbf{x})$. We furthermore use application specific metrics to evaluate the accuracy of the mean estimates.

For each application we run a set of experiments. First, we assess the sensitivity of Adam (in the context of SVI) and SVIGL *w. r. t.* to their parameters. Therefore, we first evaluate different step sizes α of Adam while fixing the number samples that are used in each iteration to approximate the KL divergence to $|\mathcal{Z}| = 50$. Note, that SVIGL does not require a step size. Having obtained an optimal step size for Adam we adapt the number of samples $|\mathcal{Z}|$, both for Adam and SVIGL. To account for slower convergence with fewer samples, we also adapt the number of iterations. In detail, for SVIGL we use 100, 200, and 400 iterations for sample set sizes of 50, 25, and 12, respectively. For Adam we use 1000, 2000, and 4000 iterations, respectively. We tune the hyperparameters of SGD in an analogous fashion. Here, our experiments showed for both applications that 4000 iterations with 12 samples and an initial step size of 10^{-6} is the best setting. Note, that we reduce the step size by a factor of ten after each third of iterations. Having tuned hyperparameters, we compare SVIGL, SVI with Adam, and SVI with SGD to a Laplace approximation as well as the MAP estimate. All experiments were conducted on an Intel Xeon E5-2650v4, 2.2 GHz, 12 cores.

We finally apply SVIGL to 3D surface reconstruction to demonstrate its benefit for applications outside the realm of computer vision.

4.5.1 *Optical flow*

In optical flow we want to recover a flow field $\mathbf{y}$ from two observed frames $\mathbf{x} = \{I_1, I_2\}$. We apply SVIGL to the EpicFlow energy of Revaud et al. (2015). The data term encourages gradient consistency between the first image and the warped second image while the prior penalizes strong horizontal or vertical flow gradients, *i. e.*

$$\begin{aligned} E(\mathbf{y}, \mathbf{x}) = \lambda_{\mathrm{D}} \sum_{l=1}^{L} \rho_{\mathrm{D}} \left(\left\| \left(\nabla \tilde{I}_2(\mathbf{y}) - \nabla I_1 \right)_l \right\|_2 \right) \\ + \lambda_{\mathrm{S}} \sum_{j=1}^{J} \sum_{l=1}^{L} \rho_{\mathrm{S}} \left(\left\| \left(\mathbf{f}_j * \mathbf{y} \right)_l \right\|_2 \right). \end{aligned} \tag{4.33}$$

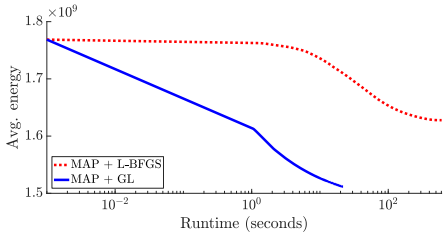
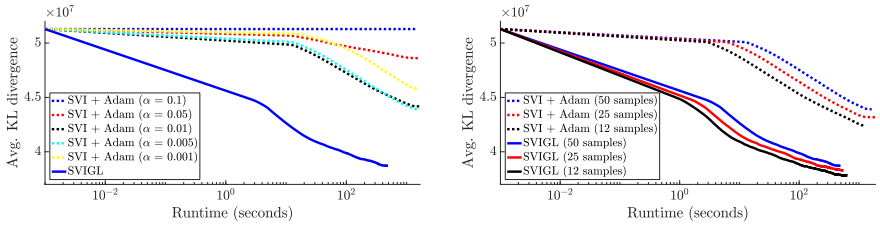


Figure 4.2. Convergence of L-BFGS and GL for MAP estimation with the optical flow energy. We show average values of the validation dataset.



(a) (b)

Figure 4.3. Convergence of SVIGL and SVI with Adam for VI on the optical flow energy. In (a) we vary the step size of Adam. In (b) we vary the number of samples and iterations for SVIGL and Adam. We show average values of the validation dataset.

Here, ∇I denotes the spatial derivatives, $\tilde{I}_2(\mathbf{y})$ is the second image warped by the flow $\mathbf{y}$, and $\mathbf{f}_1, \ldots, \mathbf{f}_J$ represent (derivative) filters.

We chose ρ_D and ρ_S as robust generalized Charbonnier functions (Barron, 2019) with associated weights λ_D, λ_S. The likelihood is linearized with a first-order Taylor approximation around the current flow.

SETUP. Following Wannenwetsch et al. (2017), we initialize the mean parameters by interpolating sparse Flowfield matches (Bailer et al., 2015) with the method of EpicFlow (Revaud et al., 2015). The variance parameter is initialized uniformly as $\sigma = 10^{-3}$. We use Bayesian optimization (Snoek et al., 2012) for finding the parameters of the Charbonnier potentials and the ratio λ_D/λ_S. Specifically, we optimize for the average end point error (AEPE) of MAP on a subset of the Sintel training set (Butler et al., 2012). We then tune the absolute scale of λ_S and λ_D on the training set such that SVIGL achieves an AEPE that is comparable to that of the MAP estimate.

RESULTS. For the following experiments we evaluation on a validation set of 104 images that are randomly chosen from Sintel training. These are strictly disjoint from the images used for parameter optimization. We start by demonstrating the benefit of gradient linearization over gradient-based optimization. Here, we evaluate MAP estimates obtained with 20 iterations of GL to those obtained with 200 iterations of L-BFGS. Figure 4.2 shows the results, averaged over the validation set,

demonstrating significantly faster optimization by GL on this highly multimodal and non-convex objective.

Let us now compare SVIGL to SVI with Adam. Instead of evaluating on full-size images, we use manually cropped 100×100 patches, thus keeping the runtime of Adam feasible. We first compare Adam with different step-sizes α to SVIGL. Figure 4.3a shows the obtained KL divergence *vs.* runtime. For the same level of KL divergence SVIGL needs two orders of magnitude less time. Additionally, while SVIGL does not require a step size Adam strongly depends on the chosen step size as suboptimal values significantly slow down convergence. We set the step size of Adam to $\alpha = 0.005$ for the remainder. Next, we compare different numbers of samples and iterations, as explained above. Figure 4.3b shows the runtime *vs.* KL divergence plots. For all settings SVIGL achieves the same level of KL divergence significantly faster than Adam. For both methods we identify a sample set size of $|\mathcal{Z}| = 12$ to be optimal.

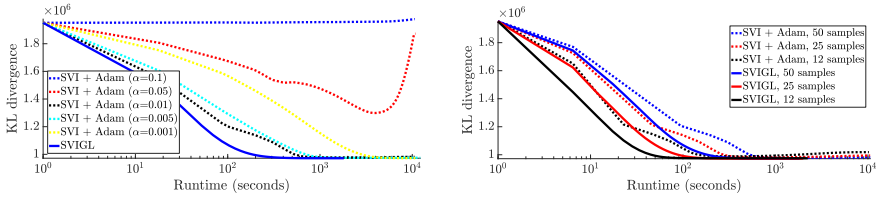
We summarize the attained KL divergence and runtime for the best hyper parameter setting of each method in Table 4.1 showing that variational approximation found by SVIGL has a significantly lower KL divergence than that obtained with SVI with Adam or SGD. As a further baseline we compute the diagonal Laplace approximation at the MAP estimate. While the Laplace approximation almost reaches the same level of KL divergence as SVIGL, it requires second-order derivatives. Moreover, the Laplace approximation leads to considerably worse results for Poisson-Gaussian denoising, *cf.* Section 4.5.2.

Finally, we compare SVIGl to MAP baselines in terms of the AEPE on the *full-size* images of Sintel test. Specifically, we evaluate against L-BFGS with 200 iterations and GL with 20 iterations. We also choose 20 iterations for SVIGL, with 50 samples each. SVIGL and GL attain an AEPE of 5.74, thus outperform L-BFGS which yields an AEPE of 5.81 Note, that Adam is too slow to run on full-sized images.

INTERPRETATION. Solving for the root of the linearized gradient (Eq. 4.28) causes an interdependent update of all variables. In other words, information can flow between all variables while a regular

Table 4.1. Achieved unnormalized KL divergence and required runtime for a fixed number of iterations for SVI on the optical flow energy. Results evaluated on 100×100 patches, cropped from our Sintel validation set.

Method	KL[$*10^7$]	runtime [s]
Initialization	5.13	–
GL + Laplace	3.83	–
SVI + SGD	4.45	**551**
SVI + Adam	4.24	1148
SVIGL *(ours)*	**3.78**	584



(a) (b)

Figure 4.4. Runtime *vs.* unnormalized KL divergence for denoising with SVIGL and SVI with Adam with different stepsize parameters α (a) and varying sizes of the sample set $|\mathcal{Z}|$ (b). Values averaged over the BSDS test set.

gradient step propagates information in a local spatial neighborhood only. We think that the performance gap between GL and SVIGL on the one hand, and gradient-based methods on the other hand is at least partly due to this global update.

UNCERTAINTY ESTIMATES. The variational approximation allows deriving per-pixel uncertainty of the flow as the marginal entropy at each pixel. We evaluate the quality of these uncertainties by comparing to the strong baseline ProbFlowFields (Wannenwetsch et al., 2017). For a fair comparison we use the same underlying discrete-continuous energy as ProbFlowFields and update the continuous variables with SVIGL while keeping the update of the discrete variables as done by Wannenwetsch et al. (2017). For further details see Appendix B. The results on the full-sized images of our validation set are shown in Table 4.2. Here, we use the metrics proposed by Wannenwetsch et al. (2017). We find that SVIGL yields competitive uncertainty estimates while just requiring a linearization of the energy gradient, leaving out the derivation of update equations needed for ProbFlowFields (Wannenwetsch et al., 2017). Figure 4.1 shows an example flow field and the associated uncertainty map.

4.5.2 *Poisson-Gaussian denoising*

Next, we apply SVIGL to the problem of removing Poisson-Gaussian noise (Section 3.3, Foi et al. (2008)). Here, it is assumed that image

Table 4.2. Comparison of uncertainty estimates obtained by SVIGL and ProbFlowFields on our Sintel validation set. We show AEPE, area under curve (AUC) of the sparsification plots, and Spearman's rank correlation coefficient. See (Wannenwetsch et al., 2017) for detail on these metrics. †Difference in AEPE due to one outlier image pair.

Method	AEPE ↑	AUC ↓	CC ↑
ProbFlowFields	3.13	0.40	0.56
SVIGL *(ours)*	3.21†	0.42	0.50

Table 4.3. Unnormalized KL divergences, PSNR values, and SSIM (Wang et al., 2004) for SVIGL and baseline methods in denoising.

Method	KL [$*10^6$]	PSNR [dB]	SSIM
Initialization	1.95	17.29	0.287
GL + Laplace	1.57	24.71	0.662
SVI + SGD	1.23	19.49	0.384
SVI + Adam	0.98	24.70	0.680
SVIGL *(ours)*	**0.97**	**24.77**	**0.693**
MAP + L-BFGS	–	23.17	0.605
MAP + GL	–	24.71	0.662

noise comes mainly from two sources that inherently affect any camera sensor. First, the Poissonian arrival process of photons hitting the pixels, and second an additive Gaussian component arising from noise in the electronics of the sensor. The Poisson distribution can be well approximated by a Gaussian (Foi et al., 2008), giving rise to a Gaussian likelihood with intensity dependent variance, *i. e.*

$$\mathbf{x}_l \sim \mathcal{N}\big(\mathbf{y}_l, \sigma(\mathbf{y}_l)^2\big) \text{ with } \sigma(\mathbf{y}_l)^2 = \beta_1 \mathbf{y}_l + \beta_2, \tag{4.34}$$

where the noise distribution is specified by the parameters β_1 and β_2. We specifically set $\beta_1 = 0.05$ and $\beta_2 = 0.0001$ in order to simulate strong noise (Poisson rate 20). Combining this likelihood with a 4-connected pairwise MRF with generalized Charbonnier potentials (Barron, 2019) as image prior leads to the energy

$$E(\mathbf{y}, \mathbf{x}) = \frac{\lambda_\mathrm{D}}{2} \sum_{l=1}^{L} \frac{(\mathbf{y}_l - \mathbf{x}_l)^2}{\sigma(\mathbf{y}_l)^2} + \lambda_\mathrm{S} \sum_{j=1}^{J} \sum_{l=1}^{L} \rho_S\Big(\big(\mathbf{f}_j * \mathbf{y}\big)_l\Big), \tag{4.35}$$

where the $\mathbf{f}_j$ denote horizontal and vertical image derivative filters. The temperature is subsumed by the weights $\lambda_\mathrm{D}, \lambda_\mathrm{S}$.

SETUP. We select the relative importance of λ_D and λ_S as well as the exponent of the robust penalty through Bayesian optimization (Snoek et al., 2012). To this end, we optimize the PSNR after 20 steps of GL on a set of 100 images from the BSDS training set (Martin et al., 2001a). We then calibrate the posterior for VI by determining the absolute scale of the weights on the training set. To synthesize noisy images for parameter tuning and testing, we apply Poisson-Gaussian noise to clean ground truth images. Afterwards, we rescale the intensities such that the ground truth lies in $[0, 1]$ and clip the noisy image to that range. For test time inference, we initialize $\boldsymbol{\mu}$ with the noisy image and $\boldsymbol{\sigma}$ as 10^{-3}.

RESULTS. We now evaluate SVIGL against SVI with Adam on the BSDS test set. In Fig. 4.4 we plot the unnormalized KL divergence against runtime for SVIGL and SVI with Adam, using varying step sizes for Adam and varying sizes of the sample set $\mathcal{Z}$ for both methods. It becomes apparent that the performance of Adam highly depends on these two parameters. Too small a step size slows down convergence, while setting it too high leads to a KL divergence inferior to the initialization. In contrast, SVIGL does not require setting a step size and converges faster than Adam with the best step size $\alpha = 0.01$. For instance, SVIGL reaches the same KL divergence as Adam in only $1/5$ of the time. When looking at the size of the sample set, we note that smaller sample sets speed up each iteration and hence lead to faster progress of the optimization. However, the solution found by Adam deteriorates after a certain number of iterations with smaller sample set sizes, while SVIGL is not affected by this issue. In summary, SVIGL yields faster convergence while being robust to the setting of nuisance parameters.

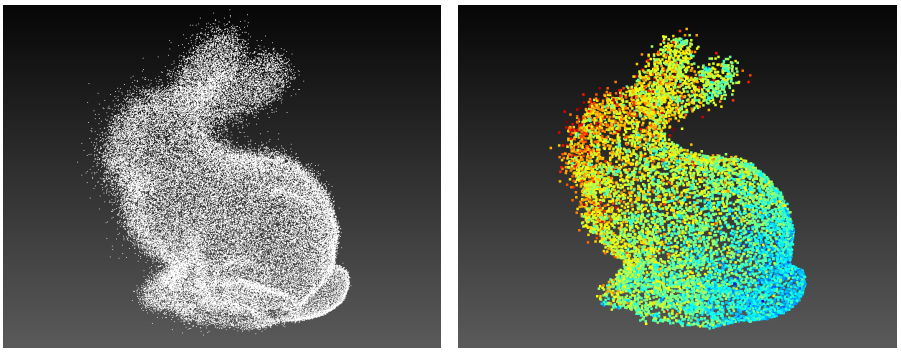
The converged solutions are evaluated in Table 4.3. SIVGL ($|\mathcal{Z}| = 50$) not only converges significantly faster than Adam ($\alpha = 0.01$, $|\mathcal{Z}| = 50$), but obtains even slightly improved solutions. SGD performs significantly worse than SVIGL and Adam. A Laplace approximation around the mode obtained with 100 iterations of GL provides a poor fit to the denoising posterior since the dependence of the variances $\sigma(\mathbf{y}_l)$ on the noise-free intensities $\mathbf{y}_l$ results in a skewed distribution. Furthermore, we see that SVIGL obtains a better solution in terms of the standard image quality metrics PSNR and SSIM (Wang et al., 2004) than the MAP estimation baselines obtained with GL and L-BFGS, *e. g.* +1.6 dB in PSNR compared to L-BFGS. In Appendix B we show denoised images obtained by SVIGL along with their uncertainty estimates.

4.5.3 *3D surface reconstruction*

In order to demonstrate that SVIGL is not limited to low-level problems in computer vision, we apply it to the task of reconstructing a smooth point cloud from noisy input data. Specifically, we use the energy of Lipman et al. (2007) given as

$$\begin{aligned} E(Y, P, C) = &\sum_{i=1}^{|Y|} \sum_{j=1}^{|P|} \|y_i - p_j\| \cdot h\big(\|c_i - p_j\|\big) \\ &- \sum_{i=1}^{|Y|} \sum_{i'=1}^{|C|} \lambda_i \|y_i - c_{i'}\| \cdot h\big(\|c_i - c_{i'}\|\big). \end{aligned} \tag{4.36}$$

Here, $p_j \in P$ denote the noisy input points; the current and the new estimate of the smoothed points are given by $c_i \in C$ and $y_i \in Y$, respectively. The contribution of each term is weighted by a Gaussian



(a) (b)

Figure 4.5. Noisy input point cloud (a) and smoothed point cloud (b); colors indicate posterior uncertainty (blue – low, red – high).

kernel $h(\cdot)$. Following Lipman et al., we use this energy in a fixed point scheme, *i. e.*

$$Y_{t+1} = \arg\min_{Y} E(Y, P, Y_t), \tag{4.37}$$

where Y_0 is an L_2 projection of the input points. In Appendix B we describe the setup in more detail.

In order to exemplify the use of SVIGL for 3D surface reconstruction, we synthesize a noisy input point cloud of the Stanford bunny by adding noise on the positions of reference points. The noise strength gradually increases from tail to face. Figure 4.5 shows both the noisy input point cloud as well as the variational approximation from SVIGL with color coded uncertainty σ. It is apparent that the uncertainty increases with input noise strength, thus reflecting the difficulty of the reconstruction task. Moreover, at points further away from the true surface, the uncertainty is generally higher, *cf.* the outliers at the ears.

4.6 CONCLUSION

Gradient linearization is a well established technique for optimizing highly multimodal and non-convex posteriors. Here, we proposed to use the same technique for stochastic variational inference leading to novel algorithm called SVIGL. Akin to gradient-based SVI it has a lightweight interface to the energy function at hand, only requiring access to a linearization of the energy gradient. Thus SVIGL is easy to apply for practitioners that want to re-use their existing energy minimization techniques for VI. We demonstrated the merits of SVIGL on the tasks of optical flow estimation, Poisson-Gaussian denoising, and 3D surface reconstruction where it yields considerably faster convergence compared to gradient-based SVI while being robust to the choice of hyper parameters. Our experiments showed that the variational approximations yield sensible uncertainty estimates that

are competitive with current state of the art that relies on tedious derivations of update equations.

5 NEURAL NEAREST NEIGHBORS NETWORKS

CONTENTS

Next, we present a novel discriminative denoising network. It is rooted in non-local methods that exploit the self-similarity of natural signals and that have been well studied, for example in image analysis and restoration. Existing approaches, however, rely on KNN matching in a fixed feature space. The main hurdle in optimizing this feature space *w. r. t.* application performance is the non-differentiability of the KNN selection rule. To overcome this, we propose a continuous deterministic relaxation of KNN selection that maintains differentiability *w. r. t.* pairwise distances, but retains the original KNN as the limit of a temperature parameter approaching zero. To exploit our relaxation, we propose the *neural nearest neighbors block* (N^3 block), a novel non-local processing layer that leverages the principle of self-similarity and can be used as building block in modern neural network architectures.[1] We show its effectiveness for the set reasoning task of correspondence classification as well as for image restoration, including image denoising and single image super-resolution, where we outperform strong CNN baselines and recent non-local models that rely on KNN selection in hand-chosen features spaces. This chapter is based on (Plötz and Roth, 2018) and extends our prior work by illustrating properties of neural nearest neighbors on a toy example in Section 5.5 and by providing further experiments on image denoising.

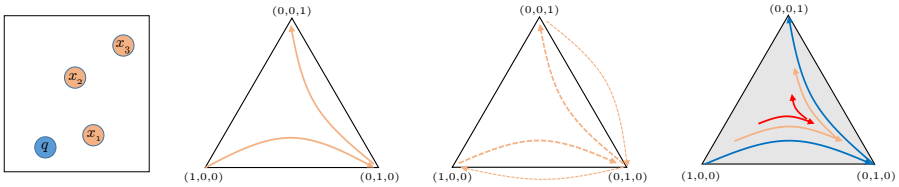
1 Code and pretrained models are available at https://github.com/visinf/n3net/.

5.1 INTRODUCTION

The ongoing surge of CNNs has revolutionized many areas of machine learning and its applications by enabling unprecedented predictive accuracy. Most network architectures focus on local processing by combining convolutional layers and element-wise operations. In order to draw upon information from a sufficiently broad context, several strategies, including dilated convolutions (Yu and Koltun, 2015) or hourglass-shaped architectures (Long et al., 2015), have been explored to increase the receptive field size. Yet, they trade off context size for localization accuracy. Hence, for many dense prediction tasks, *e. g.* in image analysis and restoration, stacking ever more convolutional blocks has remained the prevailing choice to obtain bigger receptive fields (Kim et al., 2016; Ledig et al., 2018; Mao et al., 2016; Timofte et al., 2017; Zhang et al., 2017a).

In contrast, traditional algorithms in image restoration increase the receptive field size via non-local processing, leveraging the self-similarity of natural signals. They exploit that image structures tend to re-occur within the same image (Zontak and Irani, 2011), giving rise to a strong prior for image restoration (Lotan and Irani, 2016). Hence, methods like non-local means (Buades et al., 2005a) or BM3D (Dabov et al., 2006) aggregate information across the whole image to restore a local patch. Here, matching patches are usually selected based on some hand-crafted notion of similarity, *e. g.* the Euclidean distance between patches of input intensities. Incorporating this kind of non-local processing into neural network architectures for image restoration has only very recently been considered (Lefkimmiatis, 2017; Yang and Sun, 2018). These methods replace the filtering of matched patches with a trainable network, while the feature space on which k-nearest neighbors selection is carried out is taken to be fixed. But why should we rely on a predefined matching space in an otherwise end-to-end trainable neural network architecture? In this chapter, we demonstrate that we can improve non-local processing considerably by also optimizing *w. r. t.* the feature space for matching.

The main technical challenge is imposed by the non-differentiability of the KNN selection rule. To overcome this, we make three contributions. First, we propose a continuous deterministic relaxation of the KNN rule, which allows differentiating the output *w. r. t.* pairwise distances in the input space, such as between image patches. The strength of the novel relaxation can be controlled by a temperature parameter whose gradients can be obtained as well. Second, from our relaxation we develop a novel neural network layer, called *neural nearest neighbors block* (N^3 block), which enables end-to-end trainable non-local processing based on the principle of self-similarity. Third, we demonstrate that the accuracy of image denoising and single image super-resolution (SISR) can be improved significantly by aug-



(a) Query and database (b) KNN selection (Eq. 5.2) (c) Stochastic NN (Eqs. 5.4 to 5.7) (d) Continuous NN (Eqs. 5.8 to 5.11)

Figure 5.1. *Illustration of nearest neighbors selection as paths on the simplex.* The traditional KNN rule *(b)* selects corners of the simplex deterministically based on the distance of the database items x_i to the query item q *(a)*. Stochastic neighbors selection *(c)* performs a random walk on the corners, while our proposed continuous nearest neighbors selection *(d)* relaxes the weights of the database items into the interior of the simplex and computes a deterministic path. Depending on the temperature parameter this path can interpolate between a more uniform weighting (red) and the original KNN selection (blue).

menting strong local CNN architectures with our novel N^3 block, also outperforming strong non-local baselines. Moreover, for the task of correspondence classification, we obtain significant improvements by simply augmenting a recent neural network baseline with our N^3 block, showing its effectiveness on set-valued data.

5.2 RELATED WORK

An important branch of image restoration techniques is comprised of *non-local methods* (*cf.* Section 2.2), driven by the concept of self-similarity. They rely on similar structures being more likely to encounter within an image than across images (Zontak and Irani, 2011). For denoising, the non-local means algorithm (Buades et al., 2005a) averages noisy pixels weighted by the similarity of local neighborhoods. The popular BM3D method (Dabov et al., 2006) goes beyond simple averaging by transforming the 3D stack of matching patches and employing a shrinkage function on the resulting coefficients. Such transform domain filtering is also used in other image restoration tasks, *e. g.* single image super-resolution (Cruz et al., 2018b). More recently, Yang and Sun (2018) propose to learn the domain transform and activation functions. Lefkimmiatis goes further by chaining multiple stages of trained non-local modules (Lefkimmiatis, 2017; 2018). All of these methods, however, keep the standard KNN matching in fixed feature spaces. In contrast, we propose to relax the non-differentiable KNN selection rule in order to obtain a fully end-to-end trainable non-local network.

Recently, non-local neural networks have been proposed for higher-level vision tasks such as object detection or pose estimation (Wang et al., 2018) and, with a recurrent architecture, for low-level vision

tasks (Liu et al., 2018). While also learning a feature space for distance calculation, their aggregation is restricted to a single weighted average of features, a strategy also known as *(soft) attention*. Our differentiable nearest neighbors selection generalizes this; our method can recover a single weighted average by setting k=1. As such, our novel N^3 block can potentially benefit other methods employing weighted averages, *e.g.* for visual question answering (Xu and Saenko, 2016) and more general learning tasks like modeling memory access (Graves et al., 2014) or sequence modeling (Vaswani et al., 2017). Weighted averages have also been used for building differentiable relaxations of the k-nearest neighbors *classifier* (Goldberger et al., 2006; Ren et al., 2014; Vinyals et al., 2016). Note that the crucial difference to our work is that we propose a differentiable relaxation of the KNN *selection rule* where the output is a *set* of neighbors, instead of a *single* aggregation of the labels of the neighbors. Without using relaxations, Weinberger and Saul (2009) learn the distance metric underlying KNN classification using a max-margin approach. They rely on predefined target neighbors for each query item, a restriction that we avoid.

5.3 DIFFERENTIABLE k-NEAREST NEIGHBORS

We first detail our continuous and differentiable relaxation of the KNN selection rule. Here, we will make few assumptions on the data to derive a very general result that can be used with many kinds of data, including text or sets. In the next section, we will then define a non-local neural network layer based on our relaxation. Let us start by precisely defining KNN selection. Assume that we are given a query item q, a database of candidate items $(x_i)_{i\in I}$ with indices $I = \{1, \dots, M\}$ for matching, and a distance metric $d(\cdot, \cdot)$ between pairs of items. Assuming that q is not in the database, d yields a ranking of the database items according to the distance to the query. Let $\pi_q : I \to I$ be a permutation that sorts the database items by increasing distance to q:

$$\pi_q(i) < \pi_q(i') \;\Rightarrow\; d(q, x_i) \leq d(q, x_{i'}), \quad \forall i, i' \in I. \tag{5.1}$$

The KNN of q are then given by the set of the first k items *w.r.t.* the permutation π_q

$$\mathrm{KNN}(q) \equiv \{x_i \mid \pi_q(i) \leq k\}. \tag{5.2}$$

The KNN selection rule is deterministic but not differentiable. This effectively hinders to derive gradients *w.r.t.* the distances $d(\cdot, \cdot)$. We will alleviate this problem in two steps. First, we interpret the deterministic KNN rule as a limit of a parametric family of discrete stochastic sampling processes. Second, we derive continuous relaxations for the

discrete variables, thus allowing to backpropagate gradients through the neighborhood selection while still preserving the KNN rule as a limit case.

KNN RULE AS LIMIT DISTRIBUTION. We proceed by interpreting the KNN selection rule as the limit distribution of k categorical distributions that are constructed as follows. As in Neighborhood Component Analysis (Goldberger et al., 2006), let $\mathrm{Cat}(\mathbf{w}^1 \mid \alpha^1, t)$ be a categorical distribution over the indices I of the database items, obtained by deriving logits α_i^1 from the negative distances to the query item $d(q, x_i)$, scaled with a temperature parameter t. The probability of $\mathbf{w}^1$ taking a value $i \in I$ is given by:

$$\mathbb{P}\left[\mathbf{w}^1 = i \mid \alpha^1, t\right] \equiv \mathrm{Cat}(\alpha^1, t) = \frac{\exp\left(\alpha_i^1 / t\right)}{\sum_{i' \in I} \exp\left(\alpha_{i'}^1 / t\right)} \tag{5.3}$$

$$\text{where} \quad \alpha_i^1 \equiv -d(q, x_i). \tag{5.4}$$

Here, we treat $\mathbf{w}^1$ as a one-hot coded vector and denote with $\mathbf{w}^1 = i$ that the i-th entry is set to one while the others are zero. In the limit of $t \to 0$, $\mathrm{Cat}(\mathbf{w}^1 \mid \alpha^1, t)$ will converge to a deterministic ("Dirac delta") distribution centered at the index of the database item with smallest distance to q. Thus we can regard sampling from $\mathrm{Cat}(\mathbf{w}^1 \mid \alpha^1, t)$ as a stochastic relaxation of 1-NN (Goldberger et al., 2006). We now generalize this to arbitrary k by proposing an iterative scheme to construct further conditional distributions $\mathrm{Cat}(\mathbf{w}^{j+1} \mid \alpha^{j+1}, t)$. Specifically, we compute α^{j+1} by setting the $\mathbf{w}^j$-th entry of α^j to negative infinity, thus ensuring that this index cannot be sampled again:

$$\alpha_i^{j+1} \equiv \alpha_i^j + \log(1 - \mathbf{w}_i^j) = \begin{cases} \alpha_i^j, & \text{if } \mathbf{w}^j \neq i \\ -\infty, & \text{if } \mathbf{w}^j = i. \end{cases} \tag{5.5}$$

The updated logits are used to define a new categorical distribution for the next index to be sampled:

$$\mathbb{P}\left[\mathbf{w}^{j+1} = i \mid \alpha^{j+1}, t\right] \equiv \mathrm{Cat}(\alpha^{j+1}, t) = \frac{\exp\left(\alpha_i^{j+1} / t\right)}{\sum_{i' \in I} \exp\left(\alpha_{i'}^{j+1} / t\right)}. \tag{5.6}$$

From the index vectors $\mathbf{w}^j$, we can define the *stochastic nearest neighbors* $\{X^1, \ldots, X^k\}$ of q using

$$X^j \equiv \sum_{i \in I} \mathbf{w}_i^j x_i. \tag{5.7}$$

When the temperature parameter t approaches zero, the distribution over the $\{X^1, \ldots, X^k\}$ will be a deterministic distribution centered on

the k nearest neighbors of q. Using these stochastic nearest neighbors directly within a deep neural network is problematic, since gradient estimators for expectations over discrete variables are known to suffer from high variance (Mnih and Rezende, 2016a). Hence, in the following we consider a continuous deterministic relaxation of the discrete random variables.

CONTINUOUS DETERMINISTIC RELAXATION. Our basic idea is to replace the one-hot coded weight vectors with their continuous expectations. This will yield a deterministic and continuous relaxation of the stochastic nearest neighbors that still converges to the hard KNN selection rule in the limit case of $t \to 0$. Concretely, the expectation $\bar{w}^1$ of the first index vector $\mathbf{w}^1$ is given by

$$\bar{w}_i^1 \equiv \mathbb{E}\left[\mathbf{w}_i^1 \mid \alpha^1, t\right] = \mathbb{P}\left[\mathbf{w}^1 = i \mid \alpha^1, t\right]. \tag{5.8}$$

We can now relax the update of the logits (Eq. 5.5) by using the expected weight vector instead of the discrete sample as

$$\bar{\alpha}_i^{j+1} \equiv \bar{\alpha}_i^j + \log(1 - \bar{w}_i^j) \quad \text{with} \quad \bar{\alpha}_i^1 \equiv \alpha_i^1. \tag{5.9}$$

The updated logits are then used in turn to calculate the expectation over the next index vector:

$$\bar{w}_i^{j+1} \equiv \mathbb{E}\left[\mathbf{w}_i^{j+1} \mid \bar{\alpha}^{j+1}, t\right] = \mathbb{P}\left[\mathbf{w}^{j+1} = i \mid \bar{\alpha}^{j+1}, t\right]. \tag{5.10}$$

Analogously to Eq. (5.7), we define *continuous nearest neighbors* $\{\bar{X}^1, \ldots, \bar{X}^k\}$ of q using the $\bar{w}^j$ as

$$\bar{X}^j \equiv \sum_{i \in I} \bar{w}_i^j x_i. \tag{5.11}$$

In the limit of $t \to 0$, the expectation $\bar{w}^1$ of the first sampled index vector will approach a one-hot encoding of the index of the closest neighbor. As a consequence, the logit update in Eq. (5.9) will also converge to the hard update from Eq. (5.5). By induction it follows that the other $\bar{w}^j$ will converge to a one-hot encoding of the closest indices of the j-th nearest neighbor. In summary, this means that our continuous deterministic relaxation still contains the hard KNN selection rule as a limit case.

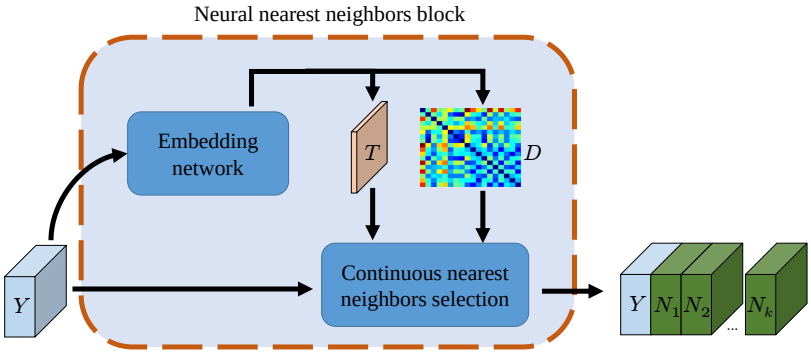
DISCUSSION. Figure 5.1 shows the relation between the deterministic KNN selection, stochastic nearest neighbors, and our proposed continuous nearest neighbors. Note that the continuous nearest neighbors are differentiable *w. r. t.* the pairwise distances as well as the temperature t. This allows making the temperature a trainable parameter. Moreover, the temperature can depend on the query item

q, thus allowing to learn for which query items it is beneficial to average more uniformly across the database items, *i.e.* by choosing a high temperature, and for which query items the continuous nearest neighbors should be close to the discrete nearest neighbors, *i.e.* by choosing a low temperature. Both cases have their justification. A more uniform averaging effectively allows to aggregate information from many neighbors at once. On the other hand, the more distinct neighbors obtained with a low temperature allow to first non-linearly process the information before eventually fusing it.

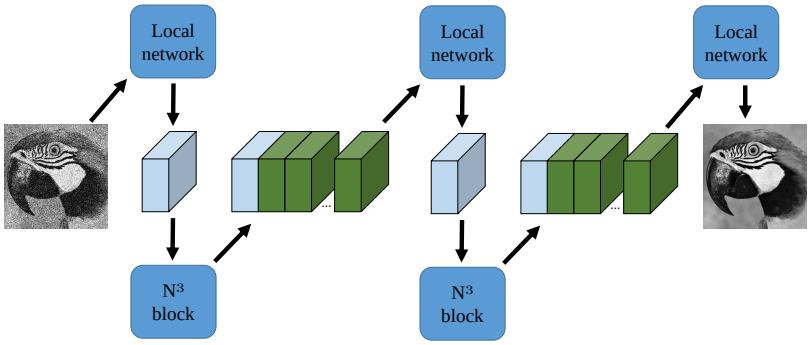
From Eq. (5.11) it becomes apparent that the continuous nearest neighbors effectively take k weighted averages over the database items. Thus, prior work such as non-local networks (Wang et al., 2018), differentiable relaxations of the KNN classifier (Vinyals et al., 2016), or soft attention-based architectures (Graves et al., 2014) can be realized as a special case of our architecture with $k = 1$. We also experimented with a continuous relaxation of the stochastic nearest neighbors based on approximating the discrete distributions with Concrete distributions (Jang et al., 2017; Maddison et al., 2017). This results in a stochastic sampling of weighted averages as opposed to our deterministic nearest neighbors. For the dense prediction tasks considered in our experiments, we found the deterministic variant to give better results, see Section 5.6.1.

5.4 NEURAL NEAREST NEIGHBORS BLOCK

In the previous section we made no assumptions about the source of query and database items. Here, we propose a new network block, called *neural nearest neighbors block* (N^3 block, Fig. 5.2a), which integrates our continuous and differentiable nearest neighbors selection into feed-forward neural networks based on the concept of *self-similarity*, *i.e.* query set and database are derived from the same features (*e.g.*, feature patches of an intermediate layer within a CNN). An N^3 block consists of two important parts. First, an embedding network takes the input and produces a feature embedding as well as temperature parameters. These are used in a second step to compute continuous nearest neighbors feature volumes that are aggregated with the input. Integrating N^3 blocks into an existing network is as easy as for a regular convolutional layer since N^3 blocks produce feature volumes of the same dimensions as the input, except for the number of feature channels which are a multiple of the input feature channels. In particular, we interleave N^3 blocks with existing local processing networks to form neural nearest neighbors networks (N^3Net) as shown in Fig. 5.2b. In the following, we take a closer look at the components of an N^3 block and their design choices.



(a) N^3 block



(b) N^3Net

Figure 5.2. *(a)* In a neural nearest neighbors (N^3) block (shaded box), an embedding network takes the output Y of a previous layer and calculates a pairwise distance matrix D between elements in Y as well as a temperature parameter (T, red feature layer) for each element. These are used to produce a stack of continuous nearest neighbors volumes $N_1, \ldots, N_k$ (green), which are then concatenated with Y. We build an N^3Net *(b)* by interleaving common local processing networks (*e. g.*, DnCNN (Zhang et al., 2017a) or VDSR (Kim et al., 2016)) with N^3 blocks.

EMBEDDING NETWORK. A first branch of the embedding network calculates a feature embedding $E = f_E(Y)$. For image data, we use CNNs to parameterize f_E; for set input we use multi-layer perceptrons. The pairwise distance matrix D can now be obtained by $D_{ij} = d(E_i, E_j)$, where E_i denotes the embedding of the i-th item and d is a differentiable distance function. We found that the Euclidean distance works well for the tasks that we consider. In practice, for each query item, we confine the set of potential neighbors to a subset of all items, *e. g.* all image patches in a certain local region. This allows our N^3 block to scale linearly in the number of items instead of quadratically. Another network branch computes a tensor $T = f_T(Y)$ containing the temperature t for each item. Note that f_E and f_T can potentially share weights to some degree. We opted for treating them as separate networks as this allows for an easier implementation.

CONTINUOUS NEAREST NEIGHBORS SELECTION. From the distance matrix D and the temperature tensor T, we compute k contin-

uous nearest neighbors feature volumes $N_1, \dots, N_k$ from the input features Y by applying Eqs. (5.8) to (5.11) to each item, *i.e.* in turn each item (*e.g.* a feature pixel or a feature point) is treated as a query item while all other items are regarded as database items. Thus, for each input item a series of k continuous nearest neighbors is formed which are then arranged spatially exactly as the corresponding input items, forming the feature volumes $N_1, \dots, N_k$. Since Y and each N_i have equal dimensionality, we could use any element-wise operation to aggregate the original features Y and the neighbors. However, a reduction at this stage would mean a very early fusion of features. Hence, we instead simply concatenate Y and the N_i along the feature dimension, which allows further network layers to learn how to fuse the information effectively in a non-linear way.

N^3 BLOCK FOR IMAGE DATA. The N^3 block described above is very generic and not limited to a certain input domain. We now describe minor technical modifications when applying the N^3 block to image data. Traditionally, non-local methods in image processing have been applied at the patch-level, *i.e.* the items to be matched consist of image patches instead of pixels. This has the advantage of using a broader local context for matching and aggregation. We follow this reasoning and first apply a strided `im2col` operation on E before calculating pairwise distances. The temperature parameter for each patch is obtained by taking the corresponding center pixel in T. Each nearest neighbor volume N_i is converted from the patch domain to the image domain by applying a `col2im` operation, where we average contributions of different patches to the same pixel.

NUMBER OF FEATURE MAPS. In order to avoid an increasing number of feature channels when inserting N^3 blocks, there are at least two convenient choices. First, one could add another 1×1 convolution after the N^3 block to reduce the number of feature channels. Second, one could decrease the number of feature channels of the input feature map such that the output features of the N^3 block again have the same number of features as the original network. For our experiments we opted for the latter as we refrained from adding an extra layer to make the N^3Net more comparable to the original network.

5.5 AN ILLUSTRATIVE TOY EXAMPLE

We now want to demonstrate benefits and limitations of the proposed neural nearest neighbor block.

THE COUNTING PROBLEM. We consider a simple but illustrative toy example where the task is to count similar objects. Specifically, let

$\mathbf{X}$ be a set of $N = 100$ items $\mathbf{x}^1, \ldots, \mathbf{x}^N$. Each item is comprised of a $D + 2$ dimensional vector

$$\mathbf{x} = [\mathrm{id}, v, n_1, \ldots, n_D], \tag{5.12}$$

where the first dimension denotes an identifier $\mathrm{id} \in \mathbb{N}$, the second dimension denotes a number $v \in \mathbb{R}$ and the remaining D dimensions $n_1, \ldots, n_D$ are given by *i. i. d.* Gaussian noise drawn from $\mathcal{N}(0, 1)$. Each item $\mathbf{x}^i$ is associated with a label $y^i \in \mathbb{N}$. Here, the label denotes the number of other items $\mathbf{x}^j \in \mathbf{X}$ that share the same identifier and whose value v^j is close to v^i:

$$y^i = \left| \left\{ \mathbf{x}^j \in \mathbf{X} : i \neq j \wedge \mathrm{id}^i = \mathrm{id}^j \wedge |v^i - v^j| < t \right\} \right|. \tag{5.13}$$

We chose a threshold of $t = 0.75$ but the actual choice is not crucial. The setting of the threshold should only ensure that the distribution of labels has a sufficient entropy. The identifiers of the 100 items in each problem instance $\mathbf{X}$ are chosen such that each integer between 0 and 9 occurs exactly ten times. It is obvious that solving this simple counting task requires two steps of reasoning. First, a suitable set of potential similar items has to be identified. Second, these selected items have to be validated and counted, which is a non-linear operation.

ARCHITECTURES. Let us now turn to different architectures for solving this problem. They all follow a basic structure, where first for each item k neighbors are selected and afterwards each item is processed individually with an MLP that has 5 layers with 100 features and ReLU non-linearities. The network outputs logits for the 10 possible values of the label. For the neighbors selection, we consider the following variants: *i)* no neighbor selection, *ii)* KNN selection ($k = 9$) where matching is done on the raw inputs, and *iii)* KNN selection ($k = 9$) where matching is done using the identifier only. Furthermore, we consider N^3 selection with varying number of selected neighbors k and a trainable linear embedding. Intuitively, the variant without neighbor selection should give a lower bound on accuracy since each item is processed completely independent of the other items. In contrast, the variant with KNN selection based on the identifier dimension should give an upper bound on the accuracy since the MLP has access to all information that is relevant for predicting the label.

RESULTS. Figure 5.3 shows the accuracy that the different architectures achieve on the counting problem for varying numbers of noise dimensions. We can make some interesting insights. First, while the KNN matching achieves almost optimal results for the noiseless case, its accuracy rapidly drops off when adding noise dimensions. Second, the accuracy of the variants with N^3 selection does degrade only slightly when adding the first noise dimension. More noise dimen-

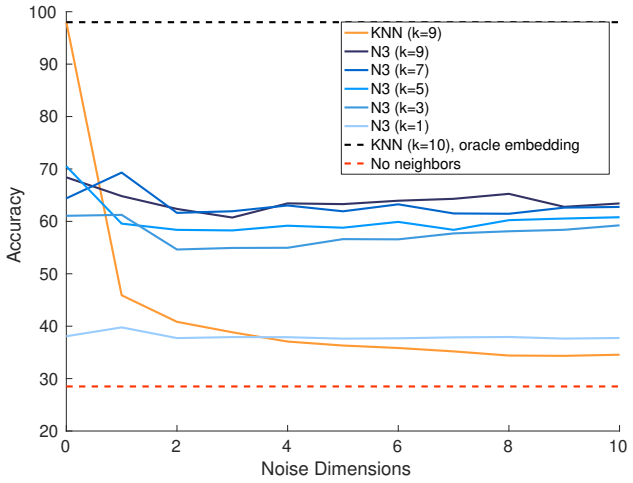


Figure 5.3. Accuracy for the counting problem.

sions do not deteriorate the accuracy further, showing the robustness of learning the embedding for matching with our differentiable neighbors selection. Third, selecting more neighbors with our N^3 block improves results. We want to emphasize that the variant with $k = 1$, *i. e.* corresponding to a single weighted average of the other items (soft attention), performs poorly with around 38% accuracy. This demonstrates that it can be highly beneficial to non-linearly process information of neighboring items instead of linearly fusing it by a single weighted average. However, when adding more neighbors the accuracy plateaus at around 63% for $k = 7$ and $k = 9$. Further analysis shows that the networks with N^3 block are able to identify a sensible embedding. But they fail in driving the temperature parameter towards zero and instead fall in a local optimum where the entropy of the weight distribution $\bar{w}^j$ (*cf.* Eq. 5.10) is still high. For the model with $k = 9$ the mean of $\max_i \bar{w}_i^1$ is just around ≈ 0.4 and the mean of $\max_i \bar{w}_i^9$ is only around 0.2. For reference, a uniform weighting across all other items in the problem instance would result in $\max_i \bar{w}_i^j \approx 0.01$ and a uniform weighting across all other items with matching identifier would yield $\max_i \bar{w}_i^j \approx 0.11$ This observation shows that the optimization problem associated with learning differentiable nearest neighbors selection is quite challenging.

5.6 EXPERIMENTS

We now analyze the properties of our novel N^3Net and show its benefits over state-of-the-art baselines. We use image denoising as our main test bed as non-local methods have been well studied there. Moreover, we evaluate on single image super-resolution and correspondence classification.

GAUSSIAN IMAGE DENOISING. We consider the task of denoising a noisy image $\mathbf{x}$, which arises by corrupting a clean image $\mathbf{y}$ with additive white Gaussian noise of standard deviation σ:

$$\mathbf{x} = \mathbf{y} + \boldsymbol{\epsilon} \quad \text{with} \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2). \tag{5.14}$$

Our baseline architecture is the DnCNN model of Zhang et al. (2017a), consisting of 16 blocks, each with a sequence of a 3×3 convolutional layer with 64 feature maps, batch normalization (Ioffe and Szegedy, 2015), and a ReLU activation function. In the end, a final 3×3 convolution is applied, the output of which is added back to the input through a global skip connection.

We use the DnCNN architecture to create our N^3Net for image denoising. Specifically, we use three DnCNNs with six blocks each, *cf.* Fig. 5.2b. The first two blocks output 8 feature maps, which are fed into a subsequent N^3 block that computes 7 neighbor volumes. The concatenated output again has a depth of 64 feature channels, matching the depth of the other intermediate blocks. The N^3 blocks extract 10×10 patches with a stride of 5. Patches are matched to other patches in a 80×80 region, yielding a total of 224 candidate patches for matching each query patch. More details on the architecture can be found in Appendix C.

TRAINING DETAILS. We follow the protocol of Zhang et al. (2017a) and use the 400 images in the train and test split of the BSD500 dataset for training. Note that these images are strictly separate from the validation images. For each epoch, we randomly crop 512 patches of size 80×80 from each training image. We use horizontal and vertical flipping as well as random rotations $\in \{0°, 90°, 180°, 270°\}$ as further data augmentation. In total, we train for 50 epochs with a batch size of 32, using the Adam optimizer (Kingma and Ba, 2015) with default parameters $\beta_1 = 0.9, \beta_2 = 0.999$ to minimize the squared error. The learning rate is initially set to 10^{-3} and exponentially decreased to 10^{-8} over the course of training. Following the publicly available implementation of DnCNN (Zhang et al., 2017a), we apply a weight decay with strength 10^{-4} to the weights of the convolution layers and the scaling of batch normalization layers.

We evaluate our full model on three different datasets: *(i)* a set of twelve commonly used benchmark images (Set12), *(ii)* the 68 images subset (Roth and Black, 2009) of the BSD500 validation set (Martin et al., 2001b), and *(iii)* the Urban100 (Huang et al., 2015) dataset, which contains images of urban scenes where repetitive patterns are abundant.

Table 5.1. PSNR and SSIM (Wang et al., 2003) on Urban100 for different architectures on gray-scale image denoising (σ=25).

	Model	Matching on	PSNR [dB]	SSIM
(i)	1 × DnCNN (d=17)	–	29.97	0.879
(ii)	1 × DnCNN (d=18)	–	29.92	0.885
(iii)	3 × DnCNN (d=6), KNN block (k=7)	noisy input	30.07	0.891
(iv)	3 × DnCNN (d=6), KNN block (k=7)	DnCNN output (d=17)	30.08	0.890
(v)	3 × DnCNN (d=6), Concrete block (k=7)	learned embedding	29.97	0.889
(ours light)	2 × DnCNN (d=6), N^3 block (k=7)	learned embedding	29.99	0.888
(ours full)	3 × DnCNN (d=6), N^3 block (k=7)	learned embedding	**30.19**	**0.892**

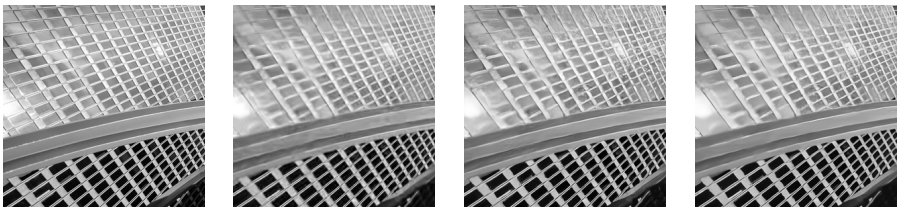
5.6.1 *Ablation studies*

We begin by discerning the effectiveness of the individual components. We compare our full N^3Net against several baselines: *(i,ii)* The baseline DnCNN network with depths 17 (default) and 18 (matching the depth of N^3Net). *(iii)* A baseline where we replace the N^3 blocks with KNN selection ($k = 7$) to obtain neighbors for each patch. Distance calculation is done on the noisy input patches. *(iv)* The same baseline as *(iii)* but where distances are calculated on denoised patches. Here we use the pretrained 17-layer DnCNN as strong denoiser. The task specific hand-chosen distance embedding for this baseline should intuitively yield more sensible nearest neighbors matches than when matching noisy input patches. *(v)* A baseline where we use Concrete distributions (Jang et al., 2017; Maddison et al., 2017) to approximately reparameterize the stochastic nearest neighbors sampling. The resulting Concrete block has an additional network for estimating the annealing parameter of the Concrete distribution.

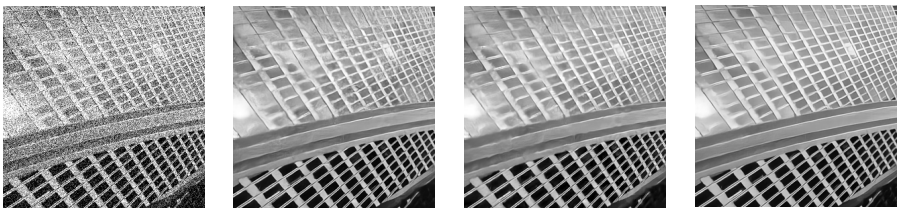
Table 5.1 shows the results on the Urban100 test set ($\sigma = 25$) from which we can infer four insights: First, the KNN baselines *(iii)* and *(iv)* improve upon the plain DnCNN model, showing that allowing the network to access non-local information is beneficial. Second, matching denoised patches (baseline *(iv)*) does not improve significantly

Table 5.2. PSNR (dB) on Urban100 for gray-scale image denoising for varying k.

	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 7$
$\sigma = 25$	30.17	30.21	30.15	**30.27**	**30.27**	30.22	30.19
$\sigma = 50$	26.76	26.81	26.78	**26.86**	26.83	26.80	26.82



(a) Clean (b) BM3D (25.21 dB) (c) FFDNet (24.92 dB) (d) NN3D (25.00 dB)



(e) Noisy (14.16 dB) (f) DnCNN (24.76 dB) (g) UNLNet (25.47 dB) (h) N^{3}Net (25.57 dB)

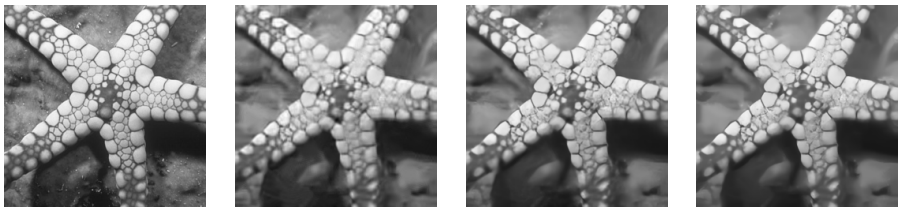
Figure 5.4. Denoising results (cropped for better display) and PSNR values on an image from Urban100 ($\sigma = 50$).

Table 5.3. PSNR (dB) for gray-scale image denoising on Set12. NLNet does not provide a model for $\sigma = 70$ and the publicly available UNLNet model was not trained for $\sigma = 70$. RED30 does not provide a model for $\sigma = 25$. Hence, we omit these results.

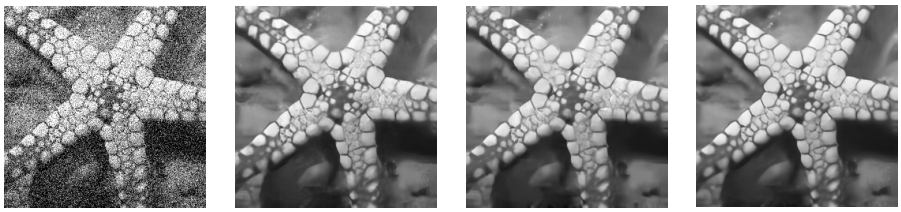
σ	DnCNN	BM3D	NLNet	UNLNet	NN3D	RED30	FFDNet	N^{3}Net
25	30.44	29.96	30.31	30.27	30.45	–	30.43	**30.55**
50	27.19	26.70	27.04	27.07	27.24	27.24	27.31	**27.43**
70	25.56	25.21	–	–	25.61	25.71	25.81	**25.90**

over matching noisy patches (baseline *(iii)*). Third, *learning* a patch embedding with our novel N^3 block shows a clear improvement over all baselines. We, moreover, evaluate a smaller version of N^{3}Net with only two DnCNN blocks of depth 6 (*ours light*). This model already outperforms the baseline DnCNN with depth 17 despite having *fewer layers* (12 *vs.* 17) and *fewer parameters* (427k *vs.* 556k). Fourth, reparameterization with Concrete distributions (baseline *(v)*) performs worse than our continuous nearest neighbors. This is probably due to the Concrete distribution introducing stochasticity into the forward pass, leading to a less stable training. Additional ablations are given in Appendix C.

DIFFERENT SETTINGS FOR k. Next, we compare N^{3}Nets with a varying number of selected neighbors. Table 5.2 shows the results on Urban100 with $\sigma \in \{25, 50\}$. We can observe that, as expected, more neighbors improve denoising results. However, the effect diminishes after roughly four neighbors and accuracy starts to deteriorate again. As we refrain from selecting optimal hyper-parameters on the test



(a) Clean (b) BM3D (24.96 dB) (c) FFDNet (25.65 dB) (d) NN3D (25.67 dB)



(e) Noisy (14.18 dB) (f) DnCNN (25.63 dB) (g) UNLNet (25.48 dB) (h) N^3Net (26.12 dB)

Figure 5.5. Denoising results (cropped for better display) and PSNR values on an image from Set12 ($\sigma = 50$).

Table 5.4. PSNR (dB) for gray-scale image denoising on BSD68. NLNet does not provide a model for $\sigma = 70$ and the publicly available UNLNet model was not trained for $\sigma = 70$. RED30 does not provide a model for $\sigma = 25$ and BSD68 is part of the RED30 training set. Hence, we omit these results.

σ	DnCNN	BM3D	NLNet	UNLNet	NN3D	RED30	FFDNet	N^3Net
25	29.23	28.56	29.03	28.99	29.19	–	29.19	**29.30**
50	26.23	25.63	26.07	26.07	26.19	–	26.29	**26.39**
70	24.85	24.46	–	–	24.89	–	25.04	**25.14**

set, we will stick to the architecture with $k = 7$ for the remaining experiments on image denoising and SISR.

LEARNED STRENGTH OF THE CONTINUOUS RELAXATION. To look into what the network has learned, we consider the maximum weight $\tilde{w}^j = \max_i \bar{w}_i^j$ (*cf.* Eq. 5.11) for the j^{th} neighbors volume. For the first N^3 block of our full network for denoising ($\sigma = 25$), we have $\tilde{w}^1 \approx 0.21$ and $\tilde{w}^7 \approx 0.11$ on average, while for the 2^{nd} block $\tilde{w}^1 \approx 0.04$ and $\tilde{w}^7 \approx 0.03$. Thus the network learned that at a lower level a "harder" N^3 selection is beneficial while for higher-level features the network tends to learn a more uniform weighting. A completely uniform weighting would correspond to $\tilde{w} = 1/224 \approx 0.004$.

RUNTIME OVERHEAD. For denoising, the runtime of our full model with N^3 increases by 3.5× compared to the baseline DnCNN model ($d = 17$). For KNN selection this overhead is 2.5×.

Table 5.5. PSNR (dB) for gray-scale image denoising on Urban100. NLNet does not provide a model for $\sigma = 70$ and the publicly available UNLNet model was not trained for $\sigma = 70$. RED30 does not provide a model for $\sigma = 25$. Hence, we omit these results.

σ	DnCNN	BM3D	NLNet	UNLNet	NN3D	RED30	FFDNet	N^{3}Net
25	29.97	29.71	29.92	29.80	30.09	–	29.92	**30.19**
50	26.28	25.95	26.15	26.14	26.47	26.32	26.52	**26.82**
70	24.36	24.27	–	–	24.53	24.63	24.87	**25.15**

5.6.2 *Comparison to the state of the art*

We compare our full N^{3}Net against state-of-the-art local denoising methods, *i. e.* the DnCNN baseline (Zhang et al., 2017a), the very deep and wide (30 layers, 128 feature channels) RED30 model (Mao et al., 2016), and the recent FFDNet (Zhang et al., 2018). Moreover, we compare against competing non-local denoisers. These include the classical BM3D (Dabov et al., 2006), which uses a hand-crafted denoising pipeline, and the state-of-the-art trainable non-local models NLNet (Lefkimmiatis, 2017) and UNLNet (Lefkimmiatis, 2018), both learning to process non-locally aggregated patches. We also compare against NN3D (Cruz et al., 2018a), which applies a non-local step on top of a pretrained network. For fair comparison, we apply a single denoising step for NN3D using our 17-layer baseline DnCNN. As a crucial difference to our proposed N^{3}Net, all of the compared non-local methods use KNN selection on a fixed feature space, thus not being able to learn an embedding for matching.

Table 5.3, Table 5.4, and Table 5.5 show the results in terms of PSNR for three different noise levels on the datasets Set12, BSD68 and Urban100, respectively. We make three important observations: First, our N^{3}Net significantly outperforms the baseline DnCNN network on all tested noise levels and all datasets. Especially for higher noise levels the margin is dramatic, *e. g.* +0.54dB ($\sigma = 50$) or +0.79dB ($\sigma = 70$) on Urban100. Even the deeper and wider RED30 model does not reach the accuracy of N^{3}Net. Second, our method is the only trainable non-local model that is able to outperform the local models DnCNN, RED30, and FFDNet. The competing models NLNet and UNLNet do not reach the accuracy of DnCNN even on Urban100, whereas our N^{3}Net even fares better than the strongest local denoiser FFDNet. Third, the post-hoc non-local step applied by NN3D is very effective on Urban100 where self-similarity can intuitively shine. However, on Set12 the gains are noticeably smaller whilst on BDS68 the non-local step can even result in degraded accuracy, *e. g.* NN3D achieves −0.04dB compared to DnCNN while N^{3}Net achieves +0.16dB for $\sigma = 50$. This highlights the importance of integrating non-local processing into an end-to-end trainable pipeline. Figure 5.4 shows denoising results for an

image from the Urban100 dataset. BM3D and UNLNet can exploit the recurrence of image structures to produce good results albeit introducing artifacts in the windows. DnCNN and FFDNet yield even more artifacts due to the limited receptive field and NN3D, as a post-processing method, cannot recover from the errors of DnCNN. In contrast, our N^{3}Net produces a significantly cleaner image where most of the facade structure is correctly restored. Figure 5.5 shows further results for an image from the Set12 dataset. Again, we can see that N^{3}Net is able to recover much finer structures compared to competing denoising methods.

5.6.3 *Real image denoising*

To further demonstrate the merits of our approach, we applied the same N^{3}Net architecture as before to the task of denoising real-world images with realistic noise. To this end, we evaluate on the recent Darmstadt Noise Dataset (Chapter 3), consisting of 50 noisy images shot with four different cameras at varying ISO levels. Realistic noise can be well explained by a Poisson-Gaussian distribution which, in turn, can be well approximated by a Gaussian distribution where the variance depends on the image intensity via a linear noise level function (*cf.* Section 3.3). We use this heteroscedastic Gaussian distribution to generate synthetic noise for training. Specifically, we use a broad range of noise level functions covering those that occur on the test images. For training, we use the 400 images of the BSDS training and test splits, 800 images of the DIV2K training set (Agustsson and Timofte, 2017), and a training split of 3793 images from the Waterloo database (Ma et al., 2017b). Before adding synthetic noise, we transform the clean RGB images $\mathbf{y}_{\text{RGB}}$ to $\mathbf{y}_{\text{RAW}}$ such that they more closely resemble images with raw intensity values:

$$\mathbf{y}_{\text{RAW}} = f_c \cdot g(\mathbf{y}_{\text{RGB}})^{f_e}, \tag{5.15}$$

$$\text{with} \quad f_c \sim \mathcal{U}(0.25, 1) \tag{5.16}$$

$$\text{and} \quad f_e \sim \mathcal{U}(1.25, 10), \tag{5.17}$$

where $g(\cdot)$ computes luminance values from RGB, the exponentiation with f_e aims at undoing compression of high image intensities, and scaling with f_c aims at undoing the effect of white balancing. Further training details can be found in Appendix C.

We input three channels to the denoising network. One corresponds to a channel of the color filter array while the other two channels contain the values β_1 and β_2 of the noise level function, respectively. We process each channel of the color filter array independently. We train both the DnCNN baseline as well as our N^{3}Net with the same training protocol and evaluate them on the benchmark website. Results are

Table 5.6. Results on the Darmstadt Noise Dataset (Chapter 3).

	Raw		sRGB	
	PSNR	SSIM	PSNR	SSIM
BM3D	46.64	0.9724	37.78	0.9308
DnCNN	47.37	0.9760	38.08	0.9357
N^3Net	**47.56**	**0.9767**	**38.32**	0.9384
TWSC	–	–	37.94	0.9403
CBDNet	–	–	38.06	**0.9421**

shown in Table 5.6. At the time of submission N^3Net sets a new state of the art for denoising raw images, outperforming DnCNN and BM3D by a significant margin. Moreover, the PSNR values, when evaluated on developed sRGB images, surpass those of the top performing methods in sRGB denoising at that time, TWSC (Xu et al., 2018a) and CBDNet (Guo et al., 2019).

Figure 5.6 shows denoising results on a real world image taken with a Sony A7R camera at a high ISO of 25600. We visually compare the result of our N^3Net to BM3D when applied either to raw image intensities or to sRGB intensities. As we can see N^3Net retains more structure than BM3D applied to raw intensities. At the same time the result of N^3Net has significantly less artifacts and residual noise than BM3D applied to sRGB intensities. Please note that our model still introduces color artifacts, *e. g.* at the lens, due to denoising all channels of the color filter array independently. This can potentially be improved by adopting a training scheme that optimizes for quality in sRGB space, *e. g.* as done in (Brooks et al., 2019).

5.6.4 *Single image super-resolution*

We now show that we can also augment recent strong CNN models for SISR with our N^3 block. We particularly consider the common task (Huang et al., 2015; Kim et al., 2016) of upsampling a low-resolution image that was obtained from a high-resolution image by bicubic downscaling. We chose the VDSR model (Kim et al., 2016) as our baseline architecture, since it is conceptually very close to the DnCNN model for image denoising. The only notable difference is that it has

Table 5.7. PSNR (dB) for single image super-resolution on Set5.

	Bicubic	SelfEx	WSD-SR	MemNet	MDSR	VDSR	N^3Net
×2	33.68	36.49	37.21	37.78	38.11	37.53	37.57
×3	30.41	32.58	33.50	34.09	34.66	33.66	33.84
×4	28.43	30.31	31.39	31.74	32.50	31.35	31.50

(a) Noisy

(b) BM3D on raw

(c) BM3D on sRGB

(d) N^{3}Net

Figure 5.6. Denoising results for a real image taken with a Sony A7R at ISO 25600.

20 layers instead of 17. We derive our N^{3}Net for SISR from the VDSR model by stacking three VDSR networks with depth 7 and inserting two N^3 blocks ($k = 7$) after the first two VDSR networks, *cf.* Fig. 5.2b. Following (Kim et al., 2016), the input to our network is the bicubicly upsampled low-resolution image and we train a single model for super-resolving images with factors 2, 3, and 4. Further details on the architecture and training protocol can be found in Appendix C. Note that we refrain from building our N^{3}Net for SISR from more recent networks, *e. g.* MemNet (Tai et al., 2017), MDSR (Lim et al., 2017), or WDnCNN (Bae et al., 2017), since they are too costly to train.

We compare our N^{3}Net against VDSR and MemNet as well as two non-local models: SelfEx (Huang et al., 2015) and the recent WSD-SR (Cruz et al., 2018b). Table 5.7 shows results on Set5 (Bevilacqua et al., 2012). Again, we can observe a consistent gain of N^{3}Net compared to the strong baseline VDSR for all super-resolution factors, *e. g.* +0.15dB for $\times 4$ super-resolution. More importantly, the other non-local methods perform inferior compared to our N^{3}Net (*e. g.* +0.36dB compared to WSD-SR for $\times 2$ super-resolution), showing that learning the match-

ing feature space is superior to relying on a hand-defined feature space. Further quantitative and visual results demonstrating the same benefits of N^3Net can be found in Appendix C.

5.6.5 *Correspondence classification*

As a third application, we look at classifying correspondences between image features from two images as either correct or incorrect. Again, we augment a baseline network with our non-local block. Specifically, we build upon the context normalization network (Yi et al., 2018), which we call CNNet in the following. The input to this network is a *set of pairs of image coordinates* of putative correspondences and the output is a probability for each of the correspondences to be correct. CNNet consists of 12 blocks, each comprised of a local fully connected layer with 128 feature channels that processes each point individually, and a context normalization and batch normalization layer that pool information across the whole point set. We augment CNNet by introducing a N^3 block after the sixth original block. As opposed to the N^3 block for the previous two tasks, where neighbors are searched only in the vicinity of a query patch, here we search for nearest neighbors among all correspondences. We want to emphasize that this is a pure *set reasoning task*. Image features are used only to determine putative correspondences while the network itself is agnostic of any image content.

For training we use the publicly available code of (Yi et al., 2018). We consider two settings: First, we train on the training set of the outdoor sequence *St. Peter* and evaluate on the test set of *St. Peter* and another outdoor sequence called *Reichstag* to test generalization. Second, we train and test on the respective sets of the indoor sequence *Brown*. Table 5.8 shows the resulting mean average precision (MAP) values at different error thresholds (for details on this metric, see (Yi et al., 2018)). We compare our N^3Net to the original CNNet and a baseline that just uses all putative correspondences for pose estimation. As can be seen, by simply inserting our N^3 block we achieve a consistent and significant gain in all considered settings, increasing MAP scores by 10% to 30%. This suggests that our N^3 block can enhance local processing networks in a wide range of applications and data domains.

5.7 CONCLUSION

Non-local methods have been well studied, *e. g.*, in image restoration. Existing approaches, however, apply KNN selection on a hand-defined feature space, which may be suboptimal for the task at hand. To overcome this limitation, we introduced the first continuous relaxation of the KNN selection rule that maintains differentiability *w. r. t.* the pairwise distances used for neighbor selection. We integrated contin-

Table 5.8. MAP scores for correspondence estimation for different error thresholds and combinations of training and testing set. Higher MAP scores are better.

Train / Test	Model	5°	10°	15°	20°	25°
St. Peter / St. Peter	No Net	0.014	0.030	0.050	0.071	0.091
	CNNet	0.271	0.379	0.460	0.522	0.570
	N^3Net	**0.316**	**0.431**	**0.514**	**0.574**	**0.619**
St. Peter / Reichstag	No Net	0.0	0.038	0.064	0.111	0.158
	CNNet	0.173	0.337	0.436	0.500	0.565
	N^3Net	**0.231**	**0.442**	**0.539**	**0.601**	**0.654**
Brown / Brown	No Net	0.054	0.110	0.182	0.232	0.274
	CNNet	0.236	0.333	0.408	0.463	0.505
	N^3Net	**0.293**	**0.391**	**0.458**	**0.510**	**0.549**

uous nearest neighbors selection into a novel network block, called N^3 block, which can be used as a general building block in neural networks. We exemplified its benefit in the context of image denoising, SISR, and correspondence classification, where we outperform state-of-the-art CNN-based methods and non-local approaches. We expect the N^3 block to also benefit end-to-end trainable architectures for other input domains, such as text or other sequence-valued data.

Part III

DENOISING MODULO IMAGES

6

JOINT DENOISING AND HDR RECONSTRUCTION FROM MULTIPLE MODULO IMAGES

CONTENTS

While the last two chapters were concerned with denoising images that are captured with a regular sensor, we will use this last technical chapter to treat denoising in the context of a novel sensor concept called modulo sensor. It is motivated by the fact that digital sensors are necessarily limited in the dynamic range they can capture. However, natural scenes often exhibit a high range of intensity values, thus exceeding the sensor capabilities. To tackle this problem, we could reconstruct an HDR image from multiple, bracketed exposures. Here, modulo sensors, introduced by Zhao et al. (2015), are an interesting alternative to regular saturating sensors as they maintain detail in bright areas of a scene. Recent multi-exposure reconstruction algorithms for the modulo sensor have shown robustness to image noise (Lang et al., 2017). However, they treat each exposure individually and do not specifically try to remove image noise, leading to suboptimal visual results. In this chapter, we propose to jointly reconstruct and denoise a series of modulo images. Therefore, we cast the reconstruction problem in a probabilistic framework and solve for the MAP estimate of the resulting posterior distribution. We show that our approach significantly improves reconstruction quality for realistic scenes.

6.1 INTRODUCTION

When taking images of realistic environments we often have to deal with a high dynamic range of the scene (Mantiuk et al., 2015), *e. g.* by viewing a dark shadowy valley right next to a sunlit mountain top. While the contrast between bright and dark makes a scene interesting from a photographer's point of view, it poses severe challenges to consumer cameras with saturating sensors. With their small dynamic range, *e. g.* 12 bit, these sensors will either saturate in bright areas or fail at resolving detail in dark areas. Hence, practitioners often resort to taking multiple LDR images at various exposure levels and fusing them afterwards to obtain a HDR reconstruction (*e. g.* Fuji Photo Film Co., Ltd., 2003; Granados et al., 2010; Grossberg and Nayar, 2003; Mantiuk et al., 2015). However, these approaches face the limitation that regular saturating sensors can not resolve detail well in bright areas, leading to artifacts.

For the sake of increasing dynamic range, alternatives to saturating sensors have been developed. For example the LSA sensor of Böhm et al. (1998) allows for locally adaptive gain control which comes at the price of an expensive hardware setup. Other alternatives trade off dynamic range for either spatial resolution (Nayar and Mitsunaga, 2000) or intensity resolution (Loose et al., 2001). Recently, Zhao et al. (2015) presented another interesting sensor concept for HDR imaging. They coined their design a *modulo camera* as each pixel site resets once it reaches its saturation level, hence effectively measuring the modulo of the image intensity at that pixel. While this allows to resolve the least significants bits of the image intensity at every pixel irregardless of the overall magnitude, the modulo sensor does not capture information on how often each pixel was reset, necessitating a dedicated reconstruction step. Hence, for the case of reconstructing from a single modulo image Zhao et al. (2015) present an approach akin to phase unwrapping methods that are well studied, *e. g.*, in radar interferometry (Goldstein et al., 1988) or magnetic resonance imaging (Chavez et al., 2002). Since single image reconstruction algorithms are based on detecting the positions of phase wrap-arounds they assume a sufficient smoothness of the underlying scene limiting their use for realistic scenes.

Hence, Zhao et al. (2015) also demonstrated reconstruction from multiple modulo images, each with a different exposure time. The benefit of using multiple modulo images compared to multiple saturating images comes at a reduced number of images necessary for the reconstruction as well as an increased bit-depth in brighter areas of the scene. Since Zhao et al. (2015) make the impractical assumption of virtually noise-free images, in prior work we investigated robust multi-image reconstruction from noisy modulo images (Lang et al., 2017). This reconstruction algorithm provides a clear benefit over the original

reconstruction method of Zhao et al. (2015) but it is still suboptimal when reconstructing from modulo images that are affected by strong noise as the reconstruction does not aim at *removing* noise. However, intuitively we should be able to aggregate information spatially and across multiple exposures to obtain a denoised reconstruction.

Hence, in this chapter we propose to jointly denoise and reconstruct the HDR radiance map from multiple bracketed and noisy exposures of a modulo camera. Like classical approaches to image restoration (*e.g.* Roth and Black, 2011) we formulate a probabilistic model of the posterior of the radiance map given the sequence of observed modulo images. The model consists of data term describing the image formation process for known noise levels, and a prior term that we chose as a simple pairwise MRF. To obtain the denoised radiance map, we utilize gradient descent to run MAP inference on the posterior. Even with the simplistic prior we achieve significantly improved accuracy compared to the original method of Zhao et al. (2015) and the robust reconstruction of Lang et al. (2017).

6.2 RELATED WORK

Due to the periodic nature of modulo images, the task of recovering the original image intensities from modulo images is tightly coupled to the task of reconstructing an image of absolute phase from a wrapped phase image, *e.g.* for remote sensing. Hence, in the following we will discuss relevant literature on phase unwrapping techniques.

SINGLE IMAGE PHASE UNWRAPPING There is a large body of work that considers the reconstruction of the absolute phase image given a single image of the wrapped phase. This process is called *phase unwrapping* and amounts to finding the fringes in the image where a phase jump, *i.e.* a wrap-around of the phase, occurs. Due to noise and strong discontinuities of the absolute phase these fringes are usually not unambiguously determined and hence we can only hope to find a good approximate solution to this under-constrained reconstruction problem. A classical technique for estimating absolute phase from two dimensional wrapped phase images is proposed by Goldstein et al. (1988). They define a local consistency criterion for phase integration paths and propose an algorithm for path selection that maximizes this local consistency. In a more contemporary fashion, Kamilov et al. (2015) define a rotation invariant energy function for single image phase unwrapping, consisting of a data term defined in the gradient domain and the Schattennorm of the Hessian as regularizer. However, they do not consider the case of noisy phase images. Bioucas-Dias and Valadao (2007) propose a pairwise energy function that is optimized via graph cuts (Boykov et al., 2001). Their approach is surprisingly similar to the method given by Zhao et al. (2015). Bioucas-Dias et al. (2008)

further extend the work of Bioucas-Dias and Valadao (2007) to joint unwrapping and denoising by first filtering the wrapped phase image such that phase jumps get preserved. The denoised phase image is then used to reconstruct the absolute phase. Reconstructing absolute phase from a single wrapped phase map is useful in domains like synthetic aperture radar imaging (Goldstein et al., 1988) or magnetic resonance imaging (Chavez et al., 2002) that deal with either dynamic scenes or moving sensors and hence do not allow for multiple temporally spaced measurements.

MULTI-FREQUENCY PHASE UNWRAPPING Similar to our approach Valadao and Bioucas-Dias (2008) define a two-frame phase unwrapping algorithm that jointly reconstructs and denoises the recovered phase. Their approach is also based on inference in a generative model. In contrast to our approach they assume homoscedastic Gaussian noise and treat both denoising and phase reconstruction as discrete optimization problems that are solved with graph cuts. We instead more faithfully model the noise as having an intensity-dependent variance and denoise in a continuous domain via gradient descent. Moreover, our continuous optimization can potentially be combined with their move making approach.

Mei et al. (2013) present an approach for multi-frequency phase unwrapping in TOF imaging, where the goal is to estimate the depth of a scene point by unwrapping a noisy phase image. Similar to us, they model the posterior of the unwrapped image with a Poisson-Gaussian model. However they assume that there are abundant samples for each of the two recorded frequencies.

Note, that techniques for fringe projection profilometry and structured light depth estimation deal with a similar problem of reconstructing absolute phase from multiple wrapped observations (Gorthi and Rastogi, 2010; Zhao et al., 1994). There, the phase is a function of the depth of a surface point whereas for modulo images the phase is a function of the intensity of the surface point. Moreover, obtaining a phase image with another frequency can be done by adapting the projected pattern while for the modulo camera the frequency depends on the exposure time and hence also affects the noise distribution of the observed wrapped intensity values. As a third difference, multi-frequency phase unwrapping techniques for depth estimation usually do not take into account spatial continuity of the reconstructed phase in order to allow for efficient inference. We refer to Zuo et al. (2016) for an extensive review of temporal phase unwrapping for fringe projection profilometry.

6.3 IMAGE FORMATION

We start by stating the image formation process that leads to an observed modulo image, following the notation of Lang et al. (2017). Let $R_{i,j} \in \mathbb{R}_0^+$, be the scene radiance illuminating a certain pixel (i, j) of the sensor. Exposing the scene with exposure time $\tau > 0$ to a theoretic digital camera of unbounded dynamic range will result in the image

$$I(\tau R) = \left\lfloor \lambda\big(\tau R + \epsilon(\tau R)\big) \right\rfloor, \tag{6.1}$$

where $\lambda > 0$ captures multiplicative factors within the process of converting incoming photons to digital values and the rounding is due to the discrete nature of the final digital values, *cf.* Eq. 3.11. The noise term $\epsilon(\tau R)$ consists of a Poisson-distributed part modeling the stochastic arrival process of photons, and a Gaussian-distributed part modeling noise within the camera electronics. As done in the literature (Foi et al., 2008) and in Chapter 3 we approximate the Poisson-Gaussian noise model with a heteroscedastic Gaussian distribution

$$\epsilon(\tau R) \sim \mathcal{N}\big(0, \sigma^2(\tau R)\big) \qquad \text{with} \qquad \sigma^2(\tau R) = \beta_1 \tau R + \beta_2. \tag{6.2}$$

As in the work of Lang et al. (2017) we assume that $\lambda = 1$ and $0 \leq R \leq 2^K$ for all pixel sites such that for an exposure time of $\tau = 1$ capturing $I(\tau R)$ implements a theoretical sensor with a large bit depth K. Let now $L < K$ be the bit depth that we can practically achieve with an existing image sensor, *e. g.* $L = 10$. Consequently, when capturing the scene with this sensor not all information can be retained. Whereas regular saturating cameras clip the intensity values of a certain pixel at the maximum value of $2^L - 1$, a modulo sensor M resets once it hits the maximal value:

$$M(\tau R) = I(\tau R) \bmod 2^L = I(\tau R) - k \cdot 2^L. \tag{6.3}$$

Thus the modulo sensor measures the L least significant bits while abandoning all other bits. Here, k denotes the number of rollovers, *i. e.* how often the sensor element has been reset during the exposure. Reconstructing the radiance map R from a modulo image M amounts to estimating the rollover map k and additionally removing the image noise ϵ. While recent works (Lang et al., 2017; Zhao et al., 2015) have only addressed the reconstruction problem in isolation, our approach jointly reconstructs and denoises the radiance map R.

6.4 GENERATIVE MODEL FOR DENOISING AND RECONSTRUCTION

We now consider the task of jointly reconstructing and denoising the radiance map R from T modulo images $M_1, \ldots, M_T$ that were taken

with exposure times $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_T = 1$, respectively. As in our previous work (Lang et al., 2017) we assume that the shortest exposure time τ_1 is chosen short enough such that no rollovers occur. We cast the reconstruction problem in a probabilistic inference framework where we want to find the MAP estimate of the posterior distribution over radiance maps given the observed modulo images (*cf.* Eq. 1.4)

$$\hat{R} = \arg\max_R \log p(R \mid M_1, \ldots, M_T) \tag{6.4}$$

with

$$\log p(R \mid M_1, \ldots, M_T) = \sum_i^T w_i \log p(M_i \mid R) + w_0 \log p(R) + \text{const}, \tag{6.5}$$

where $p(R)$ denotes a prior over radiance maps, $p(M_i \mid R)$ denotes the likelihood of observing the i-th modulo image and the weights w_i provide a trade-off between the individual likelihood terms and the prior.

6.4.1 *Likelihood*

Looking at the image formation process of modulo images (Eq. 6.3) the likelihood should ideally be chosen as a fully factorized wrapped heteroscedastic normal distribution that is moreover quantized to discrete values to amount for the rounding to integer values in Eq. 6.1. However, this comes with two drawbacks. First, a wrapped normal distribution does not admit an analytic expression for its log-density function. Second, quantization would require explicit integration over the distribution of the noise term $\epsilon(\tau_i R)$. We treat this issue by simply ignoring it since it has been shown that quantization can be approximated as uncorrelated additive uniform noise if the quantization step is sufficiently small compared to the signal amplitude (Bennett, 1948; Marco and Neuhoff, 2005) and hence does not provide useful gradients *w. r. t.* the radiance. We address the first problem by approximating the wrapped normal distribution with a heteroscedastic von Mises distribution (Mardia and Jupp, 2009) which is mathematically more convenient to use

$$\log p(M_i \mid R) = \kappa(\tau_i R) \cos\Big(c \cdot (M_i - \tau_i R)\Big) \tag{6.6}$$

$$- \log(2\pi) - \log(I_0(\kappa(\tau_i R))). \tag{6.7}$$

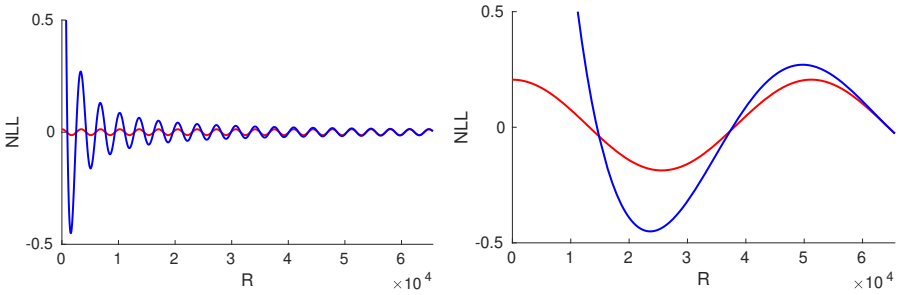


Figure 6.1. Plot of the negative log-likelihood as a function of the radiance R. We show the negative log-likelihood with the heteroscedastic noise model (blue) and a homoscedastic noise model (red) where the homoscedastic noise strength was chosen as to match the heteroscedastic noise strength at the highest radiance value. The function is shown for an observed modulo value of $M = 512$ and a bitdepth of the modulo camera of $L = 10$. The left plot shows the negative log likelihood for an exposure time of $\tau = 0.3$ while the right plot was created with $\tau = 0.02$.

Here, the factor c converts the unit of measurement from digital units to radians and is given by

$$c = \frac{2\pi}{2^L}, \tag{6.8}$$

since the observed modulo values in M_i repeat with a period of 2^L. The concentration parameter $\kappa(\tau_i R)$ of the von Mises distribution is related to the variance of the wrapped normal distribution by

$$\kappa(\tau_i R) = \frac{1}{\sigma^2(\tau_i R)} \cdot \frac{1}{c^2}. \tag{6.9}$$

The normalization constant of the von Mises distribution is given by the logarithm of I_0, *i. e.* the modified Bessel function of order 0.

Figure 6.1 shows the negative log likelihood for our heteroscedastic von Mises distribution as well as for a homoscedastic noise model as a function of the radiance R for two different exposure times, an observed modulo value $M = 512$, and a modulo camera with bitdepth $L = 10$. We can observe, that the function oscillates with a period that depends on the exposure time. Shorter exposure times yield longer periods since less rollovers can occur. Higher exposure times result in a higher frequency of the log likelihood function. Moreover, we can see that for the heteroscedastic noise model the periodic signal is modulated with a factor depending on the concentration parameter of the von Mises distribution, and hence depending on the noise strength for a certain radiance value. Since we assume signal dependent noise, smaller radiance values cause a higher amplitude of the log likelihood, while the amplitude gets smaller for higher radiance values. For the homoscedastic noise model the amplitude stays constant for all radiance values.

6.4.2 *Prior*

The likelihood of all modulo images is combined with a prior over radiance maps. In this work, we chose a simple pairwise MRF prior

$$\log p(R) = \log \mathcal{N}(R \mid 0, \sigma_p) + \sum_{(i,j)\in N} \rho(R_i - R_j), \tag{6.10}$$

with GSM potentials ρ defined for every pair of pixel indices (i, j) in the set N of neighboring pixels in a 4-neighborhood. The broad Gaussian prior $\mathcal{N}(R \mid 0, \sigma_p)$ ensures integrability. We take the GSM parameters of Schmidt et al. (2010) that were trained on natural images.

Please note, that this is a very simple prior which can certainly be improved in several ways, *e. g.* by defining higher order potentials, training the prior on a database of real radiance maps instead of natural images or by implicitly defining the prior by deep neural network based approaches (*e. g.* Heide et al., 2014; Zhang et al., 2017b). In this work we confine with demonstrating that even this simplistic pairwise prior leads to a significant improvement in reconstruction quality.

6.4.3 *Weights*

We choose the prior weight w_0 and the weights of the individual likelihood terms $w_1, \ldots, w_T$ as follows. For the prior weight we found a value of $w_0 = 0.5$ to yield good results. For the likelihood we use a uniform weighting, *i. e.* we set $w_i = 1, i = 1 \ldots T$.

6.5 INFERENCE

Optimizing for the MAP estimate (Eq. 6.4) is difficult due to the prior being non-convex and due to the periodicity of the likelihood. Hence, our inference scheme proceeds by solving a series of problems with increasing difficulty. We start by optimizing Eq. 6.4 while conditioning only on the first modulo image with the shortest exposure time,

$$\hat{R}^1 = \arg\max_R \log p(R \mid M_1). \tag{6.11}$$

Since we assume that the first exposure time is chosen such that there are no rollovers, periodicity of the likelihood does not occur. We now iteratively grow the set of modulo images in the conditioning set until we finally solve the original problem (Eq. 6.4)

$$\hat{R}^i = \arg\max_R \log p(R \mid M_1, \ldots, M_i). \tag{6.12}$$

In each stage, we initialize $\hat{R}^i$ with $\hat{R}^{i-1}$, *i. e.* the current estimate of the radiance map that was obtained from the previous stage. In the first stage we initialize by setting

$$\hat{R}^0 = M_1 / \tau_1. \tag{6.13}$$

In order to solve the optimization problem at each stage we first test if the fast pixelwise reconstruction of Lang et al. (2017) improves the negative log posterior. If it does we use the output to update the initialization of the subsequent gradient-based optimization where we employ 40 iterations of L-BFGS with line search on the negative log posterior.

Gradients of the log posterior can be obtained in a straightforward fashion. The only technical difficulty lies in calculating $\nabla_\kappa \log(I_0(\kappa))$, *i. e.* the gradient of the logarithm of the modified Bessel function, which is not available as a closed-form expression. Hence, we resort to finite differences, which is reasonable since $\log I_0$ is a smooth function with the gradient quickly approaching 1 from below as κ increases.

6.6 EXPERIMENTS

We now empirically show the effectiveness of our joint HDR denoising and reconstruction method[1]. We evaluate our reconstruction on three different datasets: *i)* the 6 radiance maps provided by Debevec and Malik (1997) (Debevec), *ii)* 8 HDR images downloaded from the HDR gallery of pfstools[2] (PFSTools) and *iii)* 24 radiance maps obtained from Fairchild's HDR photographic survey[3] (HDRPS). Each radiance map in these datasets is converted to gray scale and we crop the center 400×400 patch in order to speed up the evaluation. Each crop is furthermore scaled such we obtain a maximal value of is $2^{16} - 1$ for an exposure time of $\tau = 1$, *i. e.* we simulate imaging the scene with an ideal camera of bit depth $K = 16$. For our experiments, we simulate modulo cameras of bit depths $L \in \{10, 12, 14\}$ and with varying noise level functions. In particular, we choose noise level functions that mimic realistic settings found in our benchmark Section 3.6. Hence, we set $\beta_2 = 0.01\beta_1$ and $\beta_1 \in [10^{-4}, 10^{-1}]$, thus covering a broad range from little to very strong noise. To make results more comparable across different bit depths of the simulated modulo camera, we use the same noise level adaptation is in Lang et al. (2017). Exposure times are chosen according to the schedule presented in Lang et al. (2017).

1 Code is available at: https://github.com/tobiasploetz/modcam_denoise
2 http://pfstools.sourceforge.net/hdr_gallery.html
3 http://rit-mcsl.org/fairchild//HDR.html

6.6.1 *Ablation Study*

We perform an ablation study in order to discern the influence of the different parts of our model and inference scheme. Here, we compare different settings of our method by choosing different weightings of the prior and by evaluating different optimization schemes. Variant *(i)* uses the proposed likelihood but disregards the prior term by setting $w_0 = 0$. Variant *(ii)* directly optimizes the log posterior in a single pass by gradient-based optimization and using $\hat{R}^0$, *i. e.* the appropriately scaled first modulo image, as initialization. Variant *(iii)* does the same but uses the output of the robust reconstruction algorithm of Lang et al. (2017) as initialization. For this ablation study we set the bit depth of the modulo camera to $L = 10$ and use the noise level function $\beta_1 = 0.01$ and $\beta_2 = 10^{-4}$, thus simulating strong but realistic noise (*cf.* Section 3.6).

The PSNR and SSIM values of the reconstructed images on the Debevec dataset are shown in Table 6.1. For reference, we also show the accuracy of the robust reconstruction algorithm of Lang et al. (2017) and of the initialization of the radiance map $\hat{R}^0$. Interestingly, the prior seems to have little impact on the results as just using the likelihood achieves almost the same level of performance. This is probably due to the simplistic prior considered in this chapter and we conjecture that using a more appropriate prior will lead to significant improvements.

Regarding the optimization scheme, we observe that initialization plays an important role due to the periodic and highly non-convex likelihood. Initializing the single-pass inference with the coarsest modulo image (variant *(ii)*) performs poorly and yields almost no improvement over the radiance map initialization $\hat{R}^0$. However, results can be considerably improved by initializing with the output of (Lang et al., 2017) which can be obtained without much overhead. Finally, we observe that the iterative inference scheme *(full)* still outperforms the single-pass variants significantly, yielding an improvement of $+3.7$ dB.

6.6.2 *Comparison to State of the Art*

We now show the effectiveness of our approach when comparing to the original reconstruction algorithm of Zhao et al. (2015) and the robust reconstruction algorithm of Lang et al. (2017). A quantitative evaluation of the reconstruction results on the three datasets can be seen in Tables 6.2 to 6.4. We make the following observations: First, both our single-pass as well as our iterative reconstruction scheme outperform the baselines on all datasets and all considered settings. In some cases the margin is dramatic, *e. g.* our iterative reconstruction improves upon the robust reconstruction of Lang et al. (2017) by $+4.5$ dB on the Debevec dataset for a bit depth of $L = 10$ and a

Table 6.1. Ablation study considering different settings of our reconstruction algorithm. We show average PSNR and SSIM values of modulo image reconstruction on the Debevec dataset for bit depth $L = 10$ and a noise level function with $\beta_1 = 10^{-2}, \beta_2 = 10^{-4}$.

	Method	PSNR [dB]	SSIM
(i)	Without prior, $w_0 = 0$	50.01	0.982
(ii)	Single-Pass with $\hat{R}^0$ as initialization	37.76	0.785
(iii)	Single-Pass with (Lang et al., 2017) as initialization	46.34	0.968
(full)	Iterative full model	50.06	0.982
	Initialization $\hat{R}^0$	37.96	0.780
	Robust reconstruction (Lang et al., 2017)	45.65	0.963

noise level function with parameters $\beta_1 = 10^{-2}, \beta_2 = 10^{-4}$. This demonstrates the benefit of denoising the reconstructed images using our proposed probabilistic model. Second, our iterative approach is superior to the single-pass scheme on many settings and at the same time is never significantly worse. Especially for low bit depths of the modulo camera and high noise levels the difference between both is significant, since in these cases more images are fused than for lower noise levels and higher bit depth. We also show the accuracy of the LDR reconstruction $\hat{R}^0$ from the noisy modulo image with shortest exposure time, as well as the accuracy of a noisy HDR image that is obtained as the perfectly unwrapped modulo image with the longest exposure. Note, that this last baseline is only theoretic as it requires knowledge of the number of rollovers at each pixel. We can see, that the PSNR values of our reconstruction is always greater than the PSNR of the LDR reconstruction, demonstrating that it is beneficial to fuse the information contained in the other modulo images. At the same time, for some settings our joint reconstruction and denoising algorithm even outperforms the ideal noisy HDR image, demonstrating that we can go beyond plain reconstruction by additionally jointly denoising the radiance map.

Figure 6.2 and Fig. 6.3 show example reconstruction results for two images of the Debevec dataset. We display the reconstructions after tone mapping with the operator of Drago et al. (2003). Please note, that due to the tone mapping operator, noise is visually less severe in bright areas. We can observe that the reconstruction algorithm of Zhao et al. (2015) and the robust reconstruction of Lang et al. (2017) fail at removing the noise that is also present in a theoretical noisy HDR image. In contrast the results of our single pass reconstruction as well as the iterative reconstruction show significantly less noise in darker areas. However, there is still a considerable amount of noise

Table 6.2. Average PSNR values [dB] of modulo image reconstruction on the Debevec dataset for varying bit depths L of the modulo camera and varying noise strengths.

	$L = 10$			$L = 12$			$L = 14$		
β_1	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}
Method									
Zhao *et al.*	39.52	44.57	51.53	38.90	42.99	51.32	40.37	46.07	51.44
Lang *et al.*	45.65	68.69	80.22	48.59	63.83	73.67	46.35	56.59	66.45
Single-Pass Joint *(ours)*	46.34	71.32	**81.11**	51.40	**68.50**	**73.85**	**50.24**	**57.36**	**67.04**
Iterative Joint *(ours)*	**50.06**	**72.49**	**81.11**	**52.44**	68.48	**73.85**	49.84	**57.36**	**67.04**
Noisy LDR reconstruction	37.96	47.98	58.22	37.84	47.66	57.69	37.81	47.58	57.47
Noisy HDR reconstruction	60.32	70.28	80.22	53.92	63.83	73.67	46.76	56.59	66.45

Table 6.3. Average PSNR values [dB] of modulo image reconstruction on the PFSTools dataset for varying bit depths L of the modulo camera and varying noise strengths.

	$L = 10$			$L = 12$			$L = 14$		
β_1	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}
Method									
Zhao *et al.*	53.78	61.15	67.23	50.60	56.57	64.19	48.46	56.59	66.23
Lang *et al.*	58.08	73.87	84.08	56.69	67.26	76.79	49.69	59.37	68.72
Single-Pass Joint *(ours)*	59.77	76.46	**84.65**	59.97	**71.81**	**76.87**	53.42	**60.09**	**69.20**
Iterative Joint *(ours)*	**61.98**	**76.76**	**84.65**	**60.55**	71.78	**76.87**	**53.43**	**60.09**	**69.20**
Noisy LDR reconstruction	39.83	49.86	59.99	39.67	49.43	59.34	39.62	49.31	59.06
Noisy HDR reconstruction	64.74	74.29	84.08	57.92	67.26	76.79	49.69	59.37	68.72

left in the reconstructions and we expect that better prior models will help to drastically improve results.

6.6.3 *Noise Sensitivity*

We next analyze the reconstruction accuracy as a function of the noise characteristic and the exposure time schedule. Specifically, we probe a more fine-grained grid of noise level functions and consider four different exposure time schedules. In the first setting, we use the exposure time schedule of Lang et al. (2017). In the remaining three settings we repeat the first, second and third exposure, respectively, in order to analyze whether our reconstruction algorithm can benefit from having access to more images. Figure 6.4 shows the difference between the PSNR values of our reconstruction algorithm and the robust reconstruction of Lang et al. (2017). We can make the following observations. Repeating the first, shortest, exposure benefits mainly settings where the constant component of the noise level function, β_2, is relatively big compared to the signal-dependent component β_1. When we repeat the second and third exposures, which are longer than

(a) Clean HDR image

(b) Noisy HDR image

(c) Reconstruction with (Zhao et al., 2015)

(d) Reconstruction with (Lang et al., 2017)

(e) Our single pass reconstruction, initialized by (Lang et al., 2017)

(f) Our iterative reconstruction

Figure 6.2. Comparison of reconstruction results for an image from the Debevec dataset. A 16 bit HDR camera is to be reconstructed from 10 bit modulo images. Reconstructed images are shown after tone mapping with the operator of Drago et al. (2003).

(a) Clean HDR image

(b) Noisy HDR image

(c) Reconstruction with (Zhao et al., 2015)

(d) Reconstruction with (Lang et al., 2017)

(e) Our single pass reconstruction, initialized by (Lang et al., 2017)

(f) Our iterative reconstruction

Figure 6.3. Comparison of reconstruction results for an image from the Debevec dataset. A 16 bit HDR camera is to be reconstructed from 10 bit modulo images. Reconstructed images are shown after tone mapping with the operator of Drago et al. (2003).

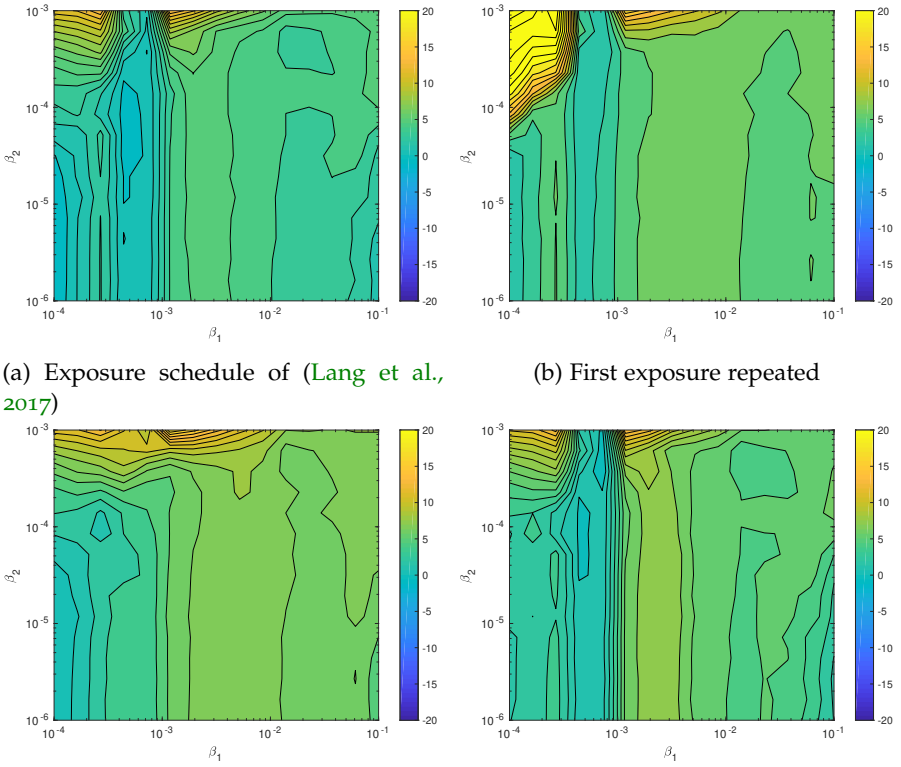
Table 6.4. Average PSNR values [dB] of modulo image reconstruction on the HDRPS dataset for varying bit depths L of the modulo camera and varying noise strengths.

	$L = 10$			$L = 12$			$L = 14$		
β_1	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}	10^{-2}	10^{-3}	10^{-4}
Method									
Zhao *et al.*	37.97	43.47	50.23	37.84	43.81	51.58	38.17	45.81	53.56
Lang *et al.*	44.26	63.37	74.21	44.76	58.05	68.11	41.33	51.49	61.46
Single-Pass Joint *(ours)*	46.14	66.28	74.98	48.07	62.51	**68.27**	**45.07**	**52.19**	**62.13**
Iterative Joint *(ours)*	**49.88**	**66.71**	**75.08**	**49.39**	**62.59**	**68.27**	44.94	**52.19**	**62.13**
Noisy LDR reconstruction	34.21	43.80	53.21	34.15	43.72	53.45	34.13	43.70	53.44
Noisy HDR reconstruction	54.52	64.49	74.39	48.21	58.17	68.14	41.74	51.50	61.46

the first, we see a more uniform improvement of the reconstruction accuracy. Especially when repeating the second exposure, there is an almost constant gain across all considered noise level functions. This demonstrates that our joint reconstruction approach can effectively leverage redundant information in the repeated exposures to improve the reconstruction while the reconstruction algorithm of Lang et al. (2017) is greedy and thus discards redundant information.

6.7 CONCLUSION

In this chapter we proposed a novel algorithm for jointly denoising and reconstructing multiple images from a modulo sensor. Our approach follows the classical line of work in image restoration by first defining a faithful generative model of the observed data which then leads to an energy minimization problem for obtaining the MAP estimate. Even though we employ a simplistic pairwise MRF prior over the reconstructed HDR image, we outperform existing approaches to modulo image reconstruction by a significant margin when images are affected by strong noise, and at the same time reconstruction accuracy does not deteriorate in low-noise settings. We conjecture that further improvements can be made by using a more sophisticated prior distribution, *e. g.* priors implicitely defined by separate denoising algorithm (Heide et al., 2014; Venkatakrishnan et al., 2013; Zhang et al., 2017b). Alternatively, discriminative learning techniques such as deep neural networks might potentially be applicable to our reconstruction problem since they have already shown promising results in the related problem of reconstructing depth from raw TOF measurements (Su et al., 2018).



(a) Exposure schedule of (Lang et al., 2017)

(b) First exposure repeated

(c) Second exposure repeated

(d) Third exposure repeated

Figure 6.4. Improvement of our joint denoising and reconstruction algorithm over the robust reconstruction of Lang et al. (2017). The plots show the difference in obtained average PSNR values on the Debevec dataset for different settings of the parameters β_1, β_2 of the noise level function. Positive values mean that our algorithm outperforms (Lang et al., 2017). In (a) the original exposure time schedule of Lang et al. (2017) is used. For (b), (c) and (d) the first, second or third exposure, respectively, is repeated two more times compared to the original exposure time schedule. It can be seen that our joint reconstruction can greatly benefit from the additional modulo images. Depending on the exposure times of the new images, the improvement affects different settings of the noise level function.

7 SUMMARY AND OUTLOOK

CONTENTS

Removing additive white Gaussian noise has been the main test bed for image denoising methods in the last decades. In this dissertation we explored multiple directions to go beyond this well established setting. We considered realistic benchmarks for denoising algorithms, novel denoising methods that fare well on practical Poisson-Gaussian noise, and finally a denoising algorithm for the modulo sensor whose noise distribution is highly non-Gaussian. In this chapter we will summarize the contributions of this thesis and give a short outlook on interesting avenues for future image denoising research.

7.1 CONTRIBUTIONS

7.1.1 *Realistic Benchmarks for Image Denoising*

The wealth of different approaches to image denoising necessitates quantitative evaluation in order to assess the accuracy of each algorithm in an objective manner. A substantial fraction of the denoising literature opts for measuring the denoising accuracy when AWGN gets synthetically added to clean images. While this is convenient, the question remains whether insights obtained on this artificial setting will generalize to denoising real noisy images.

To fill this gap, we presented a methodology for acquiring pairs of a real noisy image and corresponding ground truth in Chapter 3. Our capture setup confines with taking just two images of a scene in quick succession, a noisy image obtained with a high ISO value and a reference image obtained with a low ISO value. We demonstrated that even with a careful capture setup it is necessary to post-process the

reference image in order to obtain accurate ground truth. Crucially, this post-processing needs to take into account that real images mainly exhibit Poisson-Gaussian noise and are clipped due to sensor saturation. We postulated and experimentally validated a formation process of the noisy and reference image. Based on this we post-process the reference image by correcting small spatial translations of the camera, aligning the scale of intensities in both images, and correcting for low frequency residuals due to small changes in the ambient light. With this methodology and four diverse consumer-grade cameras we captured a set of 50 pairs of noisy and ground-truth images that comprise our novel Darmstadt Noise Dataset. While there have been earlier attempts to quantitative evaluation on real images (*e. g.* Anaya and Barbu, 2018; Nam et al., 2016), these ignore characteristics of how the images arise, thus leading to a less refined ground truth.

We used our data to benchmark recent denoising algorithms and drew the interesting conclusion that accuracy improvements made on removing synthetic AWGN on common benchmark datasets do *not* carry over to removing realistic image noise. Furthermore, the strength of image noise that is commonly considered in AWGN removal exceeds the noise strength in realistic scenarios. We also observed that discriminative methods trained on *i. i. d.* Gaussian noise do not fare well on real data. This spurred interest in developing more practical, neural network based approaches (*e. g.* Brooks et al., 2019) and we witnessed a significant improvement in accuracy on our benchmark since its release.

7.1.2 *Denoising Images from a Saturating Sensor*

We presented two novel algorithms for denoising images from conventional saturating sensors. The first algorithm is aimed at pushing the accuracy of the state of the art while the other aims at quantifying uncertainty associated with the denoised image.

COMBINING NON-LOCAL AND DISCRIMINATIVE DENOISING In recent years, discriminative approaches based on CNNs have proven to be very effective in many areas of computer vision, with image denoising being no exception (Zhang et al., 2017a). Increasing the size of the receptive field has been a successful way to improve accuracy further (Liu et al., 2017), a finding which is also supported by earlier analyses on the limits of image denoising performance (*e. g.* Levin et al., 2012) that show that classical denoising methods that are constrained to a small local context have almost saturated in terms of the MSE of the predicted denoised image.

In Chapter 5 we present a novel neural network architecture that leverages ideas of classical non-local approaches to denoising (Dabov et al., 2006) to increase the receptive field. A core contribution of this

approach is the first differentiable relaxation of the KNN selection rule. In contrast to hard KNN selection, that is employed in related non-local networks, our differentiable approximation allows to learn the feature space on which patches are matched. Moreover, it generalizes the idea of soft attention that is popularized by recent neural network architectures (Graves et al., 2014). The strength of the relaxation can be adapted through a temperature parameter and we show that hard KNN selection is obtained in the limit of the temperature approaching zero.

Based on the proposed relaxation we build a novel neural network layer, called N^3 block, and show its effectiveness for denoising images that are affected by either AWGN or realistic noise. However, the N^3 block is more general and can benefit tasks on other input domains as well, which we exemplify on the set reasoning task of correspondence classification. Here, we augmented a recent neural network baseline by simply inserting a single N^3 block and observed significant gains.

ASSESSING UNCERTAINTY FOR IMAGE DENOISING While state-of-the-art approaches like the one discussed in Chapter 5 achieve impressing accuracy, they are only able to characterize the distribution of potential clean images by a single point estimate, *i. e.* the predicted denoised image. Ideally, a denoising algorithm should accurately infer the full posterior of clean images given the noisy observation.

Instead of fully solving this challenging problem, we present a novel approach to characterize the posterior through its mean field approximation in Chapter 4. A major limitation of previous approaches to this problem is that they require tedious derivations of update equations. In contrast, we developed a stochastic variational inference algorithm that makes use of optimization via gradient linearization, a technique which is known to be superior to pure gradient based optimization for various problems in low-level computer vision. The only ingredient to our algorithm is a linearization of the gradient of the log posterior. We show that this linearization can be obtained for a class of posterior energy functions that is commonly used in low-level vision. The resulting inference scheme, called SVIGL, is fast and robust and we demonstrate it to be on par or even outperform gradient-based stochastic variational inference with a tuned Adam optimizer (Kingma and Ba, 2015).

7.1.3 *Denoising Images from a Modulo Sensor*

While denoising images taken with a regular saturating sensor is a very important problem, it is also interesting to look at other imaging techniques and to study the corresponding denoising problem. Specifically, the modulo sensor (Zhao et al., 2015) is a promising recent concept for HDR imaging. In a previous work (Lang et al., 2017) we

studied the problem of reconstructing a HDR image from a set of modulo images with different exposure times and demonstrated that this approach is able to outperform the original reconstruction algorithm of (Zhao et al., 2015).

However, the algorithm of Lang et al. (2017) is not able to actually denoise the reconstructed HDR image. Hence, in Chapter 6 we proposed to jointly reconstruct and denoise the HDR image from multiple modulo images. Following classical approaches to image restoration problems, we modeled the posterior of the clean HDR image as a combination of an observation likelihood and a prior over the clean HDR image. We obtained the denoised HDR image as the MAP estimate where optimization is done in either an iterative fashion or with a single optimization pass. We demonstrated that our method significantly improves accuracy over the reconstruction of Lang et al. (2017).

7.2 DISCUSSION AND FUTURE PERSPECTIVES

While denoising has been studied extensively for decades in the signal processing and computer vision community, there are still open problems that are worth being addressed in future work. We will shortly discuss some aspects that, as we believe, yield interesting avenues for further research.

7.2.1 *Improving Neural Network Architectures*

In terms of accuracy, denoisers based on deep neural networks (*e. g.* Zhang et al., 2017a) have been demonstrated to outperform classical methods like BM3D (Dabov et al., 2006) significantly. Although neural networks have been considered for denoising for over 10 years (Burger et al., 2012; Jain and Seung, 2009), their break through on this field was accomplished by DnCNN of Zhang et al. (2017a) and the RED models of Mao et al. (2016). These network architecture are fairly simple and we can observe some directions for further improvements.

INCREASING THE RECEPTIVE FIELD. Recent methods push the state of the art by cleverly increasing the receptive field of the networks, *e. g.* Liu et al. (2017) and Zhang et al. (2018) apply wavelet decompositions in the encoding path and their inverse in the decoding paths. This allows them to increase the receptive field in a data-independent way. In contrast our N^3Net (Chapter 5) increases the receptive field in a data-dependent way. Combining these two approaches might potentially further benefit accuracy.

INCREASING NUMBER OF FEATURES. Recent works (Liu et al., 2017) also increase the number of features compared to the earlier DnCNN model. However, this will at some point exhaust memory

during training. There are two ways to remedy this. First, we could decrease the batch size during training. Novel normalization concepts like group normalization (Wu and He, 2018) have been shown to be an adequate replacement for batch normalization (Ioffe and Szegedy, 2015), which is used by the majority of modern network architectures for denoising. In contrast to batch normalization, group normalization can also be used when the mini batch size is small or even reduced to just one data point. Second, an interesting avenue for future research is to use invertible computation layers akin to (Gomez et al., 2017; Kingma and Dhariwal, 2018). These allow to omit storing intermediate activations for the backward pass as they can be recomputed on the fly, thus lowering memory demands significantly compared to conventional non-invertible computation layers.

CONNECTIONS TO ODES AND REINFORCEMENT LEARNING. Recently, residual networks have been connected to ordinary differential equations (Haber and Ruthotto, 2017; Lu et al., 2018). Specifically, the repeated application of a residual block can be interpreted as a step of a numerical solver for some differential equation discretized in time. This leads to interesting connections to classical diffusion based models (Perona and Malik, 1990). In particular the question arises, how to chose the stopping time of the differential equation. Here, techniques from reinforcement learning and optimal control can be beneficial as recent work suggests (Yu et al., 2018; Zhang et al., 2019)

7.2.2 *Metrics*

The vast majority of denoising research uses two metrics for quantitative evaluation: The PSNR and SSIM. As the PSNR is closely related to the MSE, an optimal denoiser *w. r. t.* this metric will produce blurry and smooth results for larger noise levels. The SSIM metric alleviates this problem only to a limited extend.

Recently, Blau and Michaeli (2018) have shown in a landmark study that optimizing image-to-image metrics favors denoised images that get perceptually implausible when the noise level gets severe. Hence, they propose to use metrics that measure the distance between the distribution of denoised images and the distribution of natural images to assess the perceptual quality. However, it remains an open question how to best choose and implement the distribution distance. Using metrics based on GANs, as proposed by (Blau and Michaeli, 2018), is a popular choice in recent literature. However, care has to be taken that the research community does not put too much trust on a metric that biases the evaluation towards a certain class of models, as it might be easier for deep CNN models than for other model classes to optimize for a metric based on deep CNN features. Furthermore, metrics based on neural networks come with a lot of hyperparameters regarding

architecture and learning scheme. That being said, we fully agree with the conclusion of Blau and Michaeli (2018) that it is necessary to evaluate the distribution of denoised images in addition to the distance between ground truth and point estimates.

The next question now arises naturally: How to represent the distribution of restored images and the associated uncertainty?

7.2.3 *Uncertainty in Image Restoration*

When dealing with strong noise we have no hope to accurately pin down the original clean image with any denoising algorithm. The same holds for other types of heavily ambiguous image restoration problems like $\times 8$ super-resolution (Chen et al., 2018c) or joint deblurring and super-resolution (Xu et al., 2017b). However, as discussed in the last paragraph, optimizing for a single restored image that should be close to the ground truth, will yield some form of average over all plausible image. Since the average is not necessarily representative of the underlying distribution of clean images, we argue that it is better to try to characterize the full posterior distribution of denoised images rather than settling for a single point-estimate. We made some progress in this regard with our SVIGL algorithm in Chapter 4. However, the mean field assumption is clearly too restrictive. In the last years there has been a lot of work on powerful deep generative models, especially GANs (Arjovsky et al., 2017; Goodfellow et al., 2014) and variational auto-encoders (VAEs) (Kingma and Welling, 2014) that can be used to characterize a distribution by drawing samples. However, they lack explicit likelihood evaluations. Thus, auto-regressive models (Oord et al., 2016) or generative models using normalizing flows (Dinh et al., 2017; Kingma and Dhariwal, 2018) are promising alternatives to predict the posterior distribution of clean images. Especially, the model of Kingma and Dhariwal (2018) allows for efficient likelihood evaluation and sampling. However, the operations of normalizing flows are subject to some heavy restrictions, *i.e.* they need to be invertible and the log determinant of the Jacobian must be efficiently computable. Ideally, the inverse of the transformations should also be amenable to an efficient calculation. Instead of modeling the posterior of the restored image, Abdelhamed et al. (2019b) use normalizing flows to model the distribution of realistic noise patterns. Besides demonstrating good generative properties of their model they also use it to simulate training data for neural network based denoisers, obtaining clear accuracy gains for realistic image denoising.

7.2.4 *Practical Denoising Beyond AWGN*

Most research still focuses on removing AWGN due to its simplicity and the abundance of data that can be synthesized. However, AWGN is a

poor model for real image noise. Hence, we highlight some directions for future research focusing on practical denoising applications.

RAW IMAGE DENOISING. Our observations in Chapter 5 and the findings of Brooks et al. (2019) suggest that we can train discriminative models for raw image denoising successfully with synthetic Poisson-Gaussian noise. A key ingredient is the suitable choice of clean training images, where a common theme is to undo steps of the camera processing pipeline in order to transform sRGB intensities such that they resemble linear raw intensities. Future research should validate that this procedure generalizes beyond our benchmark data. When raw image denoising can not be done offline, *e. g.* in a software like Adobe Lightroom, but on the camera hardware directly, it will become necessary to reduce the computational burden of denoising models without sacrificing accuracy too much. Here, recent research on efficient alternatives to standard CNNs such as depthwise separable networks (Howard et al., 2017) or binarized neural networks (Courbariaux et al., 2016; Rastegari et al., 2016) could be fruitful. These works currently aim at image classification or segmentation tasks and hence there is still a need for efficient networks for image restoration. Fortunately, recent image restoration challenges, like the "PIRM challenge on perceptual image enhancement on smartphones" (Ignatov et al., 2018b) or studies like the AI benchmark (Ignatov et al., 2018a) are dedicated to assessing the quality of an algorithm not only according to the achieved accuracy but also according to the runtime and memory consumption on mobile devices.

MODULO IMAGE DENOISING. Our research in Chapter 6 was evaluated with simulated modulo images due to the simple fact that there are no real module sensors available. Although Zhao et al. (2015) presented a prototype of a modulo sensor in their work, there has not yet been a major camera manufacturer that supported this sensor concept. During private communication with representatives of a leading camera manufacturer we learned that the modulo sensor is still too far from commercial use such that they refrain from investing into the development. Hence, future research regarding the modulo sensor should focus on solving practical problems like robust reconstruction from a single image. Again, inference has to meet strict runtime and memory requirements in order to eventually fit on an embedded camera system.

Part IV

APPENDIX

A

SUPPLEMENTAL MATERIAL FOR CHAPTER 3

In this supplemental material we give a proof for $\mathcal{A}(y_n)$ and $R(y_n)$ being linearly uncorrelated. We, furthermore, give additional details on our novel heteroscedastic Tobit regression model (derivation, log-likelihood and its gradient) and highlight the importance of considering clipping of the noisy observations. Finally, we show additional results from our denoising benchmark.

A.1 LINEAR CORRELATION OF DEBIAS(IGT) AND RGT

Proposition 3. *The debiased image $\mathcal{A}(y_n)$ and the debiased residual image $R(y_n)$ are linearly uncorrelated.*

Proof. First, we note that the expectation of $R(y_n)$ given $\mathcal{A}(y_n)$ is zero

$$\begin{aligned}
&\mathbb{E}\left[R(y_n) \mid \mathcal{A}(y_n)\right] \\
&\qquad = \mathbb{E}\left[\mathcal{A}(y_n) - x_n \mid \mathcal{A}(y_n)\right] && \text{(A.1)} \\
&\qquad = \mathbb{E}\left[\mathcal{A}(y_n) \mid \mathcal{A}(y_n)\right] - \mathbb{E}\left[x_n \mid \mathcal{A}(y_n)\right] && \text{(A.2)} \\
&\qquad = \mathcal{A}(y_n) - \mathbb{E}\left[x_n \mid y_n\right] && \text{(A.3)} \\
&\qquad = 0, && \text{(A.4)}
\end{aligned}$$

where the third equality follows from the fact that $\mathcal{A}(\cdot)$ is invertible (Foi, 2009). Next, we observe that for two random variables X and Y, the expectation of X is zero if $\mathbb{E}\left[X \mid y = Y\right] = 0$ for all y:

$$\mathbb{E}\left[X\right] = \mathbb{E}_Y\left[\mathbb{E}_{X|Y}\left[X\right]\right] = \mathbb{E}_Y\left[0\right] = 0. \tag{A.5}$$

We now show that two random variables X and Y have zero covariance if $\mathbb{E}[X \mid y = Y] = 0$ for all y:

$$\begin{aligned}
\operatorname{Cov}(X, Y) &= \mathbb{E}\big[(X - \mathbb{E}X)(Y - \mathbb{E}Y)\big] && \text{(A.6)}\\
&= \mathbb{E}\big[X(Y - \mathbb{E}Y)\big] && \text{(A.7)}\\
&= \mathbb{E}[XY] - \mathbb{E}X \cdot \mathbb{E}Y && \text{(A.8)}\\
&= \mathbb{E}_Y\left[\mathbb{E}_{X|Y}[XY]\right] && \text{(A.9)}\\
&= \mathbb{E}_Y\left[Y \cdot \mathbb{E}_{X|Y}[X]\right] && \text{(A.10)}\\
&= \mathbb{E}_Y[Y \cdot 0] && \text{(A.11)}\\
&= 0. && \text{(A.12)}
\end{aligned}$$

From the definition of the linear correlation coefficient it follows that zero covariance between two random variables implies that they are linearly uncorrelated. □

A.2 HETEROSCEDASTIC TOBIT REGRESSION

We now derive the log-likelihood and its gradient of the proposed heteroscedastic Tobit regression model (Eqs. 3.24–3.26). Moreover, we detail the approximation of the noise term of Eqs. (3.22) to (3.23). For clarity, we denote $\alpha(x_r) = \boldsymbol{\alpha}^\top \mathbf{x} \doteq \tilde{x}$, where $\mathbf{x} = [x_r, 1]^\top$.

A.2.1 *Log-likelihood*

Before deriving the log-likelihood of Eq. (3.24), let us first look at the theoretical case of unclipped intensities x'_n in the high-ISO image:

$$x'_n = \tilde{x} + \epsilon_{r,n}(\tilde{x}). \tag{A.13}$$

Following Eq. (3.25), the conditional distribution of x'_n given the intensities in $\tilde{x}$ is given as a heteroscedastic Gaussian:

$$p(x'_n \mid x_r) = \mathcal{N}\big(x'_n \mid \tilde{x},\ \sigma_{r,n}(\tilde{x})\big). \tag{A.14}$$

We now consider the clipped noisy signal x_n. To derive its conditional distribution in case that x_n is clipped, we replace the Gaussian PDF with Dirac deltas weighted by the probability mass of all possible values x'_n that would be clipped to x_n. Hence, the conditional distribution

is given by a case distinction on whether x_n is unclipped, clipped from below, or from above, respectively. Precisely, we can write

$$\mathcal{T}(x_n \mid x_r) = \begin{cases} \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big), & \\ & \text{if } 0<x_n<1 \\ \delta(x_n) \cdot \int_{-\infty}^{0} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\ \mathrm{d}x'_n, & \\ & \text{if } x_n \leq 0 \\ \delta(1-x_n) \cdot \int_{1}^{\infty} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\ \mathrm{d}x'_n, & \\ & \text{if } x_n \geq 1. \end{cases} \tag{A.15}$$

It is easy to check that $\mathcal{T}(x_n \mid x_r)$ indeed is a valid probability distribution. Obviously, $\mathcal{T}(x_n \mid x_r) \geq 0$ and

$$\begin{aligned} \int_{\mathbb{R}} \mathcal{T}(x_n \mid x_r)\ \mathrm{d}x_n &= \int_{-\infty}^{0} \mathcal{T}(x_n \mid x_r)\ \mathrm{d}x_n && \text{(A.16)} \\ &\quad + \int_{0}^{1} \mathcal{T}(x_n \mid x_r)\ \mathrm{d}x_n + \int_{1}^{\infty} \mathcal{T}(x_n \mid x_r)\ \mathrm{d}x_n \\ &= \int_{-\infty}^{0} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\ \mathrm{d}x'_n && \text{(A.17)} \\ &\quad + \int_{0}^{1} \mathcal{N}\big(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\ \mathrm{d}x_n \\ &\quad + \int_{1}^{\infty} \mathcal{N}\big(x'_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\big)\ \mathrm{d}x'_n \\ &= 1. && \text{(A.18)} \end{aligned}$$

By denoting the cumulative distribution function of a standard normal distribution as $\Psi(z) = \int_{-\infty}^{z} \mathcal{N}(z' \mid 0, 1)\ \mathrm{d}z'$ and by noting that $\Psi\big(\frac{z-\mu}{\sigma}\big) = \int_{-\infty}^{z} \mathcal{N}(z' \mid \mu, \sigma)\ \mathrm{d}z'$, we can write the log-likelihood of $\mathcal{T}(x_n \mid x_r)$ up to constants as

$$\log \mathcal{T}(x_n \mid x_r) = \begin{cases} -\log \sigma_{r,n}(\tilde{x}) - \frac{(x_n-\tilde{x})^2}{2\sigma_{r,n}(\tilde{x})^2}, & \\ & \text{if } 0<x_n<1 \\ \delta(x_n) \cdot \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right), & \\ & \text{if } x_n \leq 0 \\ \delta(1-x_n) \cdot \log\left(1 - \Psi\left(\frac{1-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)\right), & \\ & \text{if } x_n \geq 1. \end{cases} \tag{A.19}$$

For constant $\sigma_{r,n}(\tilde{x}) = \sigma_{r,n}$ (*i. e.*, stationary noise) this is the log-likelihood of Tobit regression with clipping at 0 from below and at 1 from above (Tobin, 1958). In our special case, we use a non-constant link function for the standard deviation, *i. e.*

$$\sigma_{r,n}(\tilde{x}) = \sqrt{\beta_1^{r,n}\tilde{x} + \beta_2^{r,n}} \tag{A.20}$$

$$= \sqrt{(\beta_1^r + \beta_1^n)\tilde{x} + \beta_2^r + \beta_2^n} \tag{A.21}$$

in order to define our heteroscedastic Tobit regression model.

To estimate its parameters, we minimize the negative log-likelihood of all data points

$$(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}) = \underset{\boldsymbol{\alpha},\boldsymbol{\beta}^{r,n}}{\arg\min} \sum_i -\log \mathcal{T}\left(x_n^{(i)} \mid x_r^{(i)}\right). \tag{A.22}$$

A.2.2 *Log-likelihood Gradient*

It is useful to first derive the partial derivatives of terms of the form $(c-\tilde{x})/\sigma_{r,n}(\tilde{x})$ for some constant c *w. r. t.* all variables. The partial derivatives can be shown to be given as:

$$\frac{\partial (c-\tilde{x})/\sigma_{r,n}(\tilde{x})}{\partial \beta_1^{r,n}} = -\frac{1}{2}\left(c - \boldsymbol{\alpha}^\top \mathbf{y}\right) \cdot \left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-3/2} \cdot \boldsymbol{\alpha}^\top \mathbf{y} \tag{A.23}$$

$$\frac{\partial (c-\tilde{x})/\sigma_{r,n}(\tilde{x})}{\partial \beta_2^{r,n}} = -\frac{1}{2}\left(c - \boldsymbol{\alpha}^\top \mathbf{y}\right) \cdot \left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-3/2} \tag{A.24}$$

$$\frac{\partial (c-\tilde{x})/\sigma_{r,n}(\tilde{x})}{\partial \boldsymbol{\alpha}} = -\mathbf{y} \cdot \left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-1/2} - \frac{1}{2}\left(c - \boldsymbol{\alpha}^\top \mathbf{y}\right) \cdot \left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-3/2} \cdot \beta_1^{r,n}\mathbf{y}. \tag{A.25}$$

That allows to derive the partial derivatives for all three cases of the log-likelihood function. For the first case they are given as

$$\begin{aligned}\frac{\partial \log \mathcal{N}\left(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\partial \beta_1^{r,n}} = \\ -\frac{1}{2}\left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-1} \cdot \boldsymbol{\alpha}^\top \mathbf{y} \\ -\frac{x_n - \boldsymbol{\alpha}^\top \mathbf{y}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n - \tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_1^{r,n}}\end{aligned} \tag{A.26}$$

$$\begin{aligned}\frac{\partial \log \mathcal{N}\left(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\partial \beta_2^{r,n}} = \\ -\frac{1}{2}\left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-1} \\ -\frac{x_n - \boldsymbol{\alpha}^\top \mathbf{y}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n - \tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_2^{r,n}}\end{aligned} \tag{A.27}$$

$$\begin{aligned}\frac{\partial \log \mathcal{N}\left(x_n \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\partial \boldsymbol{\alpha}} = \\ -\frac{1}{2}\left(\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}\right)^{-1} \cdot \beta_1^{r,n}\mathbf{y} \\ -\frac{x_n - \boldsymbol{\alpha}^\top \mathbf{y}}{\sqrt{\beta_1^{r,n}\boldsymbol{\alpha}^\top \mathbf{y} + \beta_2^{r,n}}} \cdot \frac{\partial {}^{(x_n - \tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \boldsymbol{\alpha}}.\end{aligned} \tag{A.28}$$

To compute the last term of each equation we employ Eqs. (A.23) to (A.25). For the second case the gradient of the log-likelihood is given by

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \beta_1^{r,n}} = \frac{\mathcal{N}\left(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_1^{r,n}} \tag{A.29}$$

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \beta_2^{r,n}} = \frac{\mathcal{N}\left(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \beta_2^{r,n}} \tag{A.30}$$

$$\frac{\partial \log \Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)}{\partial \boldsymbol{\alpha}} = \frac{\mathcal{N}\left(0 \mid \tilde{x}, \sigma_{r,n}(\tilde{x})\right)}{\Psi\left(\frac{-\tilde{x}}{\sigma_{r,n}(\tilde{x})}\right)} \cdot \frac{\partial {}^{(-\tilde{x})}/_{\sigma_{r,n}(\tilde{x})}}{\partial \boldsymbol{\alpha}}, \tag{A.31}$$

again employing Eqs. (A.23) to (A.25). The third case works analogously. In practice, we optimize for $\boldsymbol{\beta}' = \log \boldsymbol{\beta}^{r,n}$ to ensure that $\boldsymbol{\beta}^{r,n}$ is positive. Furthermore, we exclude pixels near image edges (Foi et al., 2008) from the regression and truncate the log-likelihood to be robust to outliers, *i. e.* we set the gradients to zero for pixels with $\log \mathcal{T}(x_n^i \mid x_r^i) < -10$.

When estimating the $\boldsymbol{\alpha}$ parameter for the image pairs in our dataset, we use previously recorded noise parameters $\boldsymbol{\beta}$. These were obtained from running our full Tobit regression on controlled images showing a color checker, see Fig. A.1.

A.2.3 *Approximation of Noise Term*

Here, we quantify the error that is induced by approximating the noise term in Eqs. (3.22) to (3.23). Specifically, we approximate the variance of the Gaussian noise by

$$\alpha_1^2 \left(\beta_1^r x_r + \beta_2^r\right) \approx \beta_1^r (\alpha_1 x_r + \alpha_2) + \beta_2^r. \tag{A.32}$$

Obviously, the left-hand side would converge to the right-hand side as $\alpha_1 \to 1$ and $\alpha_2 \to 0$, if the ISO value and exposure time were set with perfect accuracy. In practice, however, this is not the case. We now evaluate the practical impact of our approximation. With denoting $\beta(x_r) = \beta_1 x_r + \beta_2$, let

$$\sigma(x_r) = \sqrt{\alpha_1^2 \beta^r(x_r) + \beta^n(\alpha(x_r))} \tag{A.33}$$

be the true noise level function and

$$\hat{\sigma}(x_r) = \sqrt{\beta^{r,n}(\alpha(x_r))} \tag{A.34}$$

be the approximated noise level function. We compute the normalized root mean squared error Φ (Mäkitalo and Foi, 2014) between the true and the approximated noise level function, assuming a uniform distribution over pixel intensities

$$\Phi(\sigma, \hat{\sigma}) = \int_0^1 \frac{\left(\sigma(x_r) - \hat{\sigma}(x_r)\right)^2}{\sigma(x_r)} \, \mathrm{d}x_r. \tag{A.35}$$

The average normalized RMSE on our dataset is $1.4 \cdot 10^{-4}$, meaning that on average approximating the noise standard deviation introduces a relative error of 0.014%. This is insignificant compared to the overall estimation accuracy (see Section 3.6).

A.3 BIAS FROM CLIPPING

Figure A.2 plots the noise-free image intensities y_n against the average of clipped noisy observations x_n for the noise level function of the Nexus 6P at ISO 6400. We can see that the mean of the clipped observations strongly deviates from the true noise-free intensities near the

clipping boundaries, also see (Foi, 2009). Due to this bias introduced by clipping the signal, we can not recover the noise free signal by simply averaging noisy observations spatially or temporally. Hence, we perform the smoothing operation of our low-frequency residual correction in the debiased domain, *cf.* Eqs. (3.28) and (3.29).

A.4 SIMULATION OF POISSON-GAUSSIAN NOISE

For our experiments on synthetic data (Section 3.6) we apply Poisson-Gaussian noise to the noise-free images (Eq. 3.30). To demonstrate that Eq. (3.30) is sensible let x'_n again be the unclipped noisy signal. According to the heteroscedastic Gaussian noise model, *cf.* Eqs. (3.12) and (3.13), the mean and variance of x'_n are given by

$$\mathbb{E}(x'_n) = y_n, \tag{A.36}$$

$$\mathrm{Var}(x'_n) = \beta_1^n y_n + \beta_2^n. \tag{A.37}$$

Let now z_n be the unclipped simulated noisy signal of Eq. (3.30):

$$z_n \sim \beta_1^n \cdot \mathcal{P}(y_n/\beta_1^n) + \mathcal{N}(0, \sqrt{\beta_2^n}) \tag{A.38}$$

According to the properties of the Poisson distribution, the mean and variance of z_n are given by

$$\mathbb{E}(z_n) = \beta_1^n \frac{y_n}{\beta_1^n} + 0 = y_n, \tag{A.39}$$

$$\mathrm{Var}(z_n) = (\beta_1^n)^2 \frac{y_n}{\beta_1^n} + \beta_2^n = \beta_1^n y_n + \beta_2^n. \tag{A.40}$$

We can see that the two first moments of x'_n and z_n match and hence z_n provides a good simulation of the noise as given by the noise level function $\boldsymbol{\beta}^n$. The same holds for the simulation of the reference image.

A.5 ADDITIONAL RESULTS

Finally, we give a few more results obtained on our novel DND benchmark dataset. First, Fig. A.3 shows a histogram of the PSNR values of the crops of the noisy test images in linear raw space. As we can see, our dataset covers a wide range of noise levels for the noisy images, hence allowing to benchmark denoising algorithms across many different situations. Note that the mean PSNR of the noisy images (39.38 dB) is significantly below the PSNR of the reference images (52.76 dB, from the estimated noise level function). Consequently, the

ground truth accuracy of our benchmark far exceeds the performance of state-of-the-art denoising techniques (*cf.* Table 3.4), thus providing significant headroom even for future improvement in denoising techniques. Figure A.4 shows denoising results aggregated for different noise levels. The top-performing methods overall achieve consistent results across almost all noise levels. We can furthermore observe that NCSR has severe problems in denoising images affected by weak intensity-dependent noise. When applying the variance stabilizing transformation, NCSR shows a more competitive performance. For MLP we observe that performance on RAW denoising peaks for $\hat{\sigma}$ close to the noise level used for training, *i. e.* $\sigma_{\text{train}} \approx 10^{-1.41}$. For removing noise with a different noise level, MLP does not generalize well.

Finally, Figures A.5 and A.6 show denoising results of the tested algorithms for one crop of two different images in our database. The results were obtained from denoising raw intensities after the variance stabilizing transformation. We display the denoised images in linear raw space (red channel only) . We can see that many methods oversmooth fine structures (*e. g.*, MLP and FoE), while TNRD undersmoothes and fails to remove a significant part of the noise.

Figure A.1. Test scene used for noise parameter calibration.

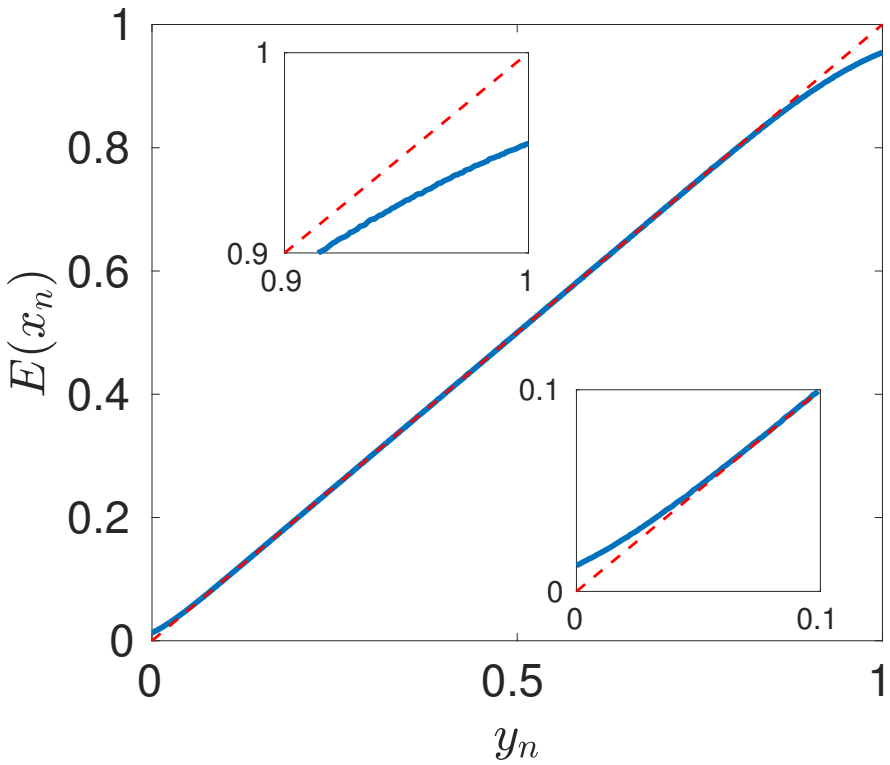


Figure A.2. Noise-free intensities (red dashed line) *vs.* mean of clipped noisy intensities (blue solid line).

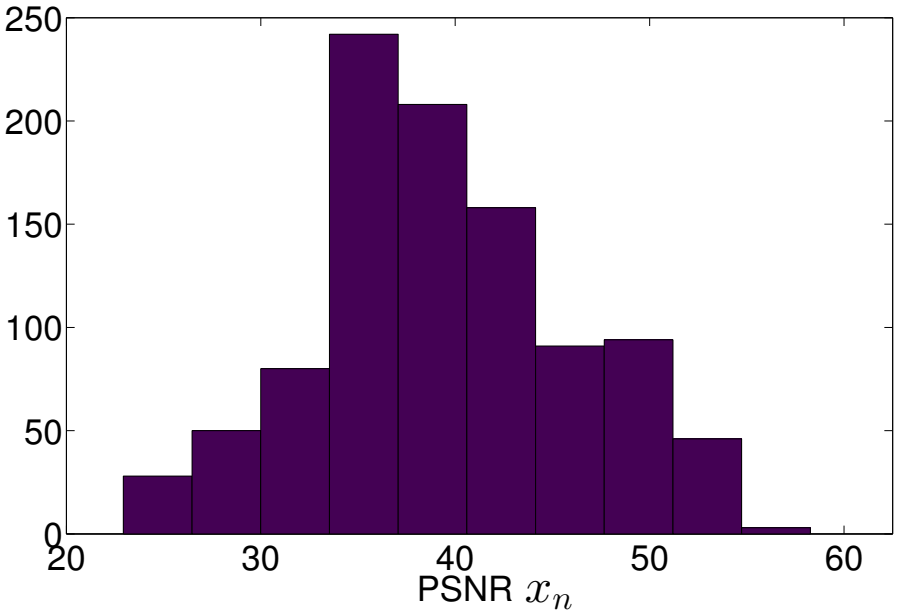
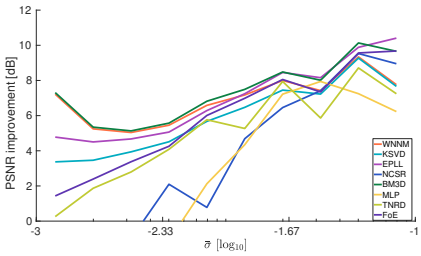
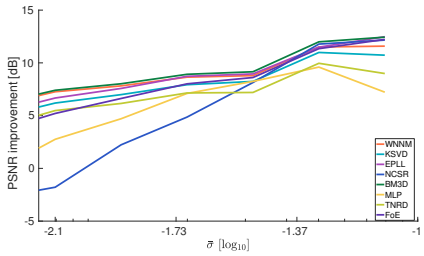


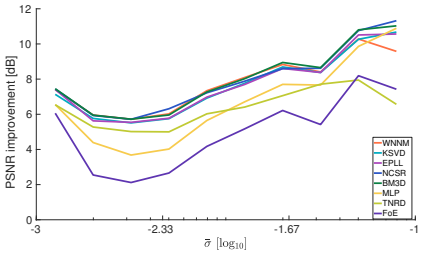
Figure A.3. Histogram of PSNR values (in dB) of the crops of the noisy test images.



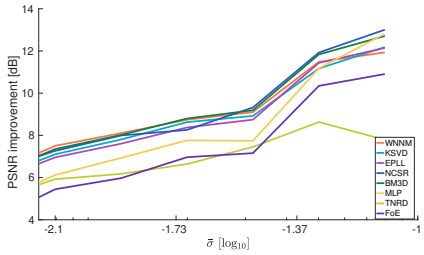
(a) Denoising raw pixels, evaluating in RAW space.



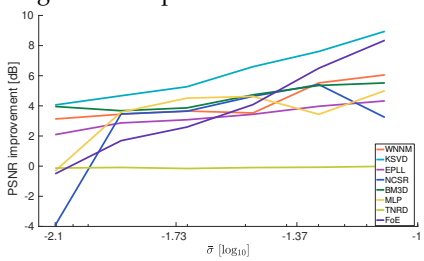
(b) Denoising raw pixels, evaluating in sRGB space.



(c) Denoising raw pixels after VST, evaluating in RAW space.

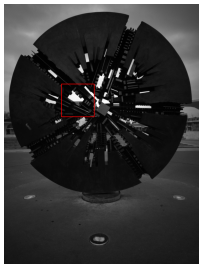


(d) Denoising raw pixels after VST, evaluating in sRGB space.

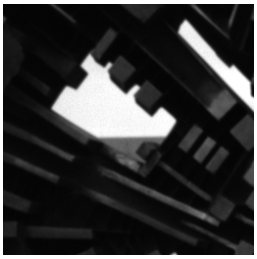


(e) Denoising in sRGB space, evaluating in sRGB space.

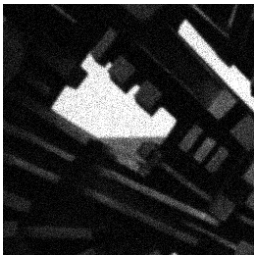
Figure A.4. Denoising performance by noise level $\bar{\sigma}$.



(a) Full-sized reference



(b) Reference



(c) Noisy (26.59 dB)

(d) WNNM (37.02 dB)

(e) KSVD (35.87 dB)

(f) EPLL (35.50 dB)

(g) NCSR (37.80 dB)

(h) BM3D (36.81 dB)

(i) MLP (37.61 dB)

(j) TNRD (32.38 dB)

(k) FoE (32.39 dB)

Figure A.5. Example denoising result (red channel only) with PSNR values, displayed in linear raw space.

(a) Full-sized reference

(b) Reference

(c) Noisy (36.34 dB)

(d) WNNM (45.46 dB)

(e) KSVD (45.04 dB)

(f) EPLL (45.54 dB)

(g) NCSR (45.51 dB)

(h) BM3D (45.78 dB)

(i) MLP (44.58 dB)

(j) TNRD (40.44 dB)

(k) FoE (42.50 dB)

Figure A.6. Example denoising result (red channel only) with PSNR values, displayed in linear raw space. Intensities of crops are uniformly scaled for better display.

B

SUPPLEMENTAL MATERIAL FOR CHAPTER 4

Here, we give an interpretation of SVIGL as preconditioned gradient-descent with a special preconditioner and derive the linearized gradient for the optical flow and Poisson-Gaussian denoising energies. Moreover, we proof Proposition 2, show results for searching hyper parameters of SVI with SGD, provide more detail on the comparison with ProbFlowFields (*cf.* Table 4.2) and the 3D surface reconstruction experiment, and give visual results for Poisson-Gaussian denoising.

B.1 SVIGL AS PRECONDITIONED GRADIENT DESCENT

Here, we give an interpretation of the SVIGL update step (Eq. 4.28) as an iteration of preconditioned gradient descent. To simplify notation let $\mathbf{A}_\theta \equiv \mathbf{A}_\theta\big(\boldsymbol{\theta}^{(t)}\big)$ and $\mathbf{b}_\theta \equiv \mathbf{b}_\theta\big(\boldsymbol{\theta}^{(t)}\big)$. Similar to, *e.g.*, Nikolova and Chan (2007), we have

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= -\mathbf{A}_\theta^{-1}\mathbf{b}_\theta && \text{(B.1)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_\theta^{-1}\mathbf{b}_\theta - \boldsymbol{\theta}^{(t)} && \text{(B.2)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_\theta^{-1}\Big(\mathbf{b}_\theta + \mathbf{A}_\theta\boldsymbol{\theta}^{(t)}\Big) && \text{(B.3)}\\
&= \boldsymbol{\theta}^{(t)} - \mathbf{A}_\theta^{-1}\nabla_\theta \operatorname{KL}\left(q \,||\, p\right). && \text{(B.4)}
\end{aligned}$$

Therefore, the SVIGL update relates to gradient descent with preconditioner $P = \mathbf{A}_\theta^{-1}$. This also enables us to add a step size parameter α to SVIGL

$$\begin{aligned}
\boldsymbol{\theta}^{(t+1)} &= \boldsymbol{\theta}^{(t)} - \alpha\mathbf{A}_\theta^{-1}\nabla_\theta \operatorname{KL}\left(q \,||\, p\right) && \text{(B.5)}\\
&= \boldsymbol{\theta}^{(t)} - \alpha\mathbf{A}_\theta^{-1}\Big(\mathbf{b}_\theta + \mathbf{A}_\theta\boldsymbol{\theta}^{(t)}\Big) && \text{(B.6)}\\
&= (1-\alpha)\boldsymbol{\theta}^{(t)} + \alpha\hat{\boldsymbol{\theta}}^{(t+1)}, && \text{(B.7)}
\end{aligned}$$

with $\hat{\boldsymbol{\theta}}^{(t+1)} = -\mathbf{A}_\theta^{-1}\mathbf{b}_\theta$ denoting the full SVIGL estimate (Eq. 4.28). In practice, we found SVIGL is rather insensitive to the choice of the step size parameter and hence simply use $\alpha = 1$.

B.2 LINEARIZED GRADIENTS

Here, we derive linearized gradients for the Poisson-Gaussian denoising energy and a, for brevity, simplified version of the optical flow energy, *i. e.* where we have replaced the gradient constancy data term with a brightness constancy data term. Note, that the derivation for the more complex energy is analogous.

B.2.1 *Optical flow*

We first write the simplified optical flow energy of Section 4.5.1 with the data term linearized by a first-order Taylor expansion around $\mathbf{y}_l^0$, *i. e.*

$$\begin{aligned} E(\mathbf{y}, \mathbf{x}) =& \lambda_{\mathrm{D}} \sum_{l=1}^{L} \rho_{\mathrm{D}} \left(I_{t,l} + \begin{pmatrix} I_{x,l} \\ \\ I_{y,l} \end{pmatrix}^{\mathrm{T}} \left(\mathbf{y}_l - \mathbf{y}_l^0 \right) \right) \\ &+ \lambda_{\mathrm{S}} \sum_{j=1}^{J} \sum_{l=1}^{L} \rho_{\mathrm{S}} \left(\left\| \left(\mathbf{f}_j * \mathbf{y} \right)_l \right\|_2 \right) \qquad \text{(B.8)} \\ =& \lambda_{\mathrm{D}} E_{\mathrm{D}}(\mathbf{y}, \mathbf{x}) + \lambda_{\mathrm{S}} E_{\mathrm{S}}(\mathbf{y}), \qquad \text{(B.9)} \end{aligned}$$

with $I_{t,l} = I_2\left(l + \mathbf{y}_l^0\right) - I_1\left(l\right)$, $\begin{pmatrix} I_{x,l} \\ \\ I_{y,l} \end{pmatrix} = \nabla I_2\left(l + \mathbf{y}_l^0\right)$.

DATA TERM. We now derive a linearized gradient for the data term. We start by noting that

$$\begin{aligned} \nabla_{\mathbf{y}_l} E_{\mathrm{D}}(\mathbf{y}, \mathbf{x}) =& \nabla_{\mathbf{y}_l} \rho_{\mathrm{D}} \left(I_{t,l} + \begin{pmatrix} I_{x,l} \\ \\ I_{y,l} \end{pmatrix}^{\mathrm{T}} \left(\mathbf{y}_l - \mathbf{y}_l^0 \right) \right) \qquad \text{(B.10)} \\ =& \rho_{\mathrm{D}}' \left(I_{t,l} + \begin{pmatrix} I_{x,l} \\ \\ I_{y,l} \end{pmatrix}^{\mathrm{T}} \left(\mathbf{y}_l - \mathbf{y}_l^0 \right) \right) \cdot \begin{pmatrix} I_{x,l} \\ \\ I_{y,l} \end{pmatrix}. \qquad \text{(B.11)} \end{aligned}$$

We now write the derivative of the generalized Charbonnier $\rho_{\mathrm{D}}(\cdot)$ (Barron, 2019) as:

$$\rho'_{\mathrm{D}}(x) = \frac{x}{c^2}\left(\frac{(x/c)^2}{\max(1, 2-a)} + 1\right)^{(a/2-1)} \tag{B.12}$$

$$\equiv \tilde{\rho}_{\mathrm{D}}(x)\ x. \tag{B.13}$$

With Eqs. (B.11) and (B.13), we get

$$\begin{aligned}
&\nabla_{\mathbf{y}_l} E_{\mathrm{D}}(\mathbf{y}, \mathbf{x}) = \\
&= \tilde{\rho}_{\mathrm{D}}\left(I_{t,l} + \begin{pmatrix} I_{x,l} \\ I_{y,l} \end{pmatrix}^{\mathrm{T}} \left(\mathbf{y}_l - \mathbf{y}_l^0\right)\right) \\
&\quad \cdot \left(\begin{pmatrix} I_{x,l} I_{t,l} \\ I_{y,l} I_{t,l} \end{pmatrix} + \begin{pmatrix} I_{x,l}^2 & I_{x,l} I_{y,l} \\ I_{x,l} I_{y,l} & I_{y,l}^2 \end{pmatrix} \left(\mathbf{y}_l - \mathbf{y}_l^0\right)\right).
\end{aligned} \tag{B.14}$$

From the last expression (Eq. B.14) we can identify a linearization of the data term gradient as

$$\nabla_{\mathbf{y}} E_{\mathrm{D}}(\mathbf{y}, \mathbf{x}) = \mathbf{A}_{\mathbf{y}}^{\mathrm{D}}(\mathbf{y})\mathbf{y} + \mathbf{b}_{\mathbf{y}}^{\mathrm{D}}(\mathbf{y}), \tag{B.15}$$

with

$$\mathbf{A}_{\mathbf{y}}^{\mathrm{D}}(\mathbf{y}) = \begin{pmatrix} \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_x^2\right) & \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_x I_y\right) \\ \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_x I_y\right) & \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_y^2\right) \end{pmatrix} \tag{B.16}$$

and

$$\mathbf{b}_{\mathbf{y}}^{\mathrm{D}}(\mathbf{y}) = \begin{pmatrix} \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_x I_t\right)\mathbf{1} \\ \mathbf{D}\left(\tilde{\rho}_{\mathrm{D}} \cdot I_y I_t\right)\mathbf{1} \end{pmatrix} - \mathbf{A}_{\mathbf{y}}(\mathbf{y})\mathbf{y}^0. \tag{B.17}$$

Here, $\mathbf{y} = \left(x_1^{(1)}, \ldots, x_L^{(1)}, x_1^{(2)}, \ldots, x_L^{(2)}\right)^{\mathrm{T}}$ is the stacked vector of the horizontal and vertical components of the flow. $\mathbf{D}(\cdot)$ is the short-hand notation for $\mathrm{diag}\{\cdot\}$,*i. e.* the diagonal matrix from the argument vector. Products are applied element-wise.

SMOOTHNESS TERM. We now consider the smoothness term. We first rewrite the convolution $\mathbf{f}_j * \mathbf{y}$ as a matrix-vector product $\mathbf{F}_j \cdot \mathbf{y}$, with $\mathbf{F}_j$ being the convolution matrix associated to $\mathbf{f}_j$ and $\mathbf{y}$ being the vectorized flow. We can now write the gradient of the smoothness term E_S as:

$$\nabla_{\mathbf{y}} E_S(\mathbf{y}) = \nabla_{\mathbf{y}} \sum_{j=1}^{J} \sum_{l=1}^{L} \rho_S\Big((\mathbf{F}_j \mathbf{y})_l \Big) \tag{B.18}$$

$$= \sum_{j=1}^{J} \mathbf{F}_j^T \rho_S'(\mathbf{F}_j \mathbf{y}). \tag{B.19}$$

Rewriting the derivative ρ_S' of the generalized Charbonnier as in Eq. (B.13), we get

$$\sum_{j=1}^{J} \mathbf{F}_j^T \rho_S'(\mathbf{F}_j \mathbf{y}) = \sum_{j=1}^{J} \mathbf{F}_j^T \, \mathbf{D}\Big(\tilde{\rho}_S(\mathbf{F}_j \mathbf{y}) \Big) \, \mathbf{F}_j \mathbf{y} \tag{B.20}$$

$$= \left(\sum_{j=1}^{J} \mathbf{F}_j^T \, \mathbf{D}\Big(\tilde{\rho}_S(\mathbf{F}_j \mathbf{y}) \Big) \, \mathbf{F}_j \right) \mathbf{y} \tag{B.21}$$

$$\equiv \mathbf{A}_{\mathbf{y}}^{S}(\mathbf{y}) \mathbf{y}. \tag{B.22}$$

COMPLETE LINEARIZED GRADIENT. We now combine the linearization of the data term gradient (Eq. B.16) and smoothness term gradient (Eq. B.22) to get a linearization of the full energy gradient as

$$\nabla_{\mathbf{y}} E(\mathbf{y}, \mathbf{x}) = \lambda_D \nabla_{\mathbf{y}} E_D(\mathbf{y}, \mathbf{x}) + \lambda_S \nabla_{\mathbf{y}} E_S(\mathbf{y}) \tag{B.23}$$

$$= \Big(\lambda_D \mathbf{A}_{\mathbf{y}}^{D}(\mathbf{y}) + \lambda_S \mathbf{A}_{\mathbf{y}}^{S}(\mathbf{y}) \Big) \mathbf{y} + \lambda_D \mathbf{b}_{\mathbf{y}}^{D} \tag{B.24}$$

$$\equiv \mathbf{A}_{\mathbf{y}}(\mathbf{y}) \mathbf{y} + \mathbf{b}_{\mathbf{y}}. \tag{B.25}$$

B.2.2 *Poisson-Gaussian denoising*

Let us first recap the energy function for Poisson-Gaussian denoising:

$$E(\mathbf{y},\mathbf{x}) = \frac{\lambda_\mathrm{D}}{2}\sum_{l=1}^{L}\frac{(\mathbf{y}_l-\mathbf{x}_l)^2}{\sigma(\mathbf{y}_l)^2} + \lambda_\mathrm{S}\sum_{j=1}^{J}\sum_{l=1}^{L}\rho_\mathrm{S}\Big((\mathbf{f}_j * \mathbf{y})_l\Big), \tag{B.26}$$

$$= \lambda_\mathrm{D} E_\mathrm{D}(\mathbf{y},\mathbf{x}) + \lambda_\mathrm{S} E_\mathrm{S}(\mathbf{y}), \tag{B.27}$$

where

$$\sigma(\mathbf{y}_l)^2 = \beta_1 \mathbf{y}_l + \beta_2. \tag{B.28}$$

We will derive the linearized gradients for the data term E_D and the smoothness term E_S separately.

DATA TERM. The gradient of the data term is given as

$$\nabla_\mathbf{y} E_\mathrm{D}(\mathbf{y},\mathbf{x})$$

$$= \frac{(\mathbf{y}-\mathbf{x})}{\sigma(\mathbf{y})^2} - \frac{\beta_1(\mathbf{y}-\mathbf{x})^2}{2\sigma(\mathbf{y})^4} \tag{B.29}$$

$$= \frac{\mathbf{y}}{\sigma(\mathbf{y})^2} - \frac{\mathbf{x}}{\sigma(\mathbf{y})^2} - \frac{\beta_1\mathbf{y}^2}{2\sigma(\mathbf{y})^4} + \frac{\beta_1\mathbf{y}\mathbf{x}}{\sigma(\mathbf{y})^4} - \frac{\beta_1\mathbf{x}^2}{2\sigma(\mathbf{y})^4} \tag{B.30}$$

$$= \mathbf{y}\left(\frac{1}{\sigma(\mathbf{y})^2} - \frac{\beta_1\mathbf{y}}{2\sigma(\mathbf{y})^4} + \frac{\beta_1\mathbf{x}}{\sigma(\mathbf{y})^4}\right) - \left(\frac{\mathbf{x}}{\sigma(\mathbf{y})^2} + \frac{\beta_1\mathbf{x}^2}{2\sigma(\mathbf{y})^4}\right), \tag{B.31}$$

where all operations are element-wise. The linearized gradient of the data term can then be obtained as

$$\mathbf{A}_\mathbf{y}^\mathrm{D}(\mathbf{y}) = \mathbf{D}\left(\frac{1}{\sigma(\mathbf{y})^2} - \frac{\beta_1\mathbf{y}}{2\sigma(\mathbf{y})^4} + \frac{\beta_1\mathbf{x}}{\sigma(\mathbf{y})^4}\right) \tag{B.32}$$

$$\mathbf{b}_\mathbf{y}^\mathrm{D}(\mathbf{y}) = -\left(\frac{\mathbf{x}}{\sigma(\mathbf{y})^2} + \frac{\beta_1\mathbf{x}^2}{2\sigma(\mathbf{y})^4}\right). \tag{B.33}$$

SMOOTHNESS TERM. For the smoothness term we can re-use the linearized gradient derived in Eq. (B.22).

COMPLETE LINEARIZED GRADIENT. We can now put the results of Eqs. (B.22), (B.32) and (B.33) together to obtain a linearized gradient of the energy for Poisson-Gaussian denoising, *cf.* Eqs. (B.23) to (B.25).

B.3 PROOF OF PROPOSITION 2

We now proof Proposition 2. Let us first reiterate the proposition statement.

Proposition. *An energy function can be linearized with a positive semi-definite matrix* $\mathbf{A}_\mathbf{y}$ *if it is composed of a sum of energy terms* $\rho_i(\mathbf{w}_i)$ *that fulfill the following conditions:*

1. *Each penalty function* $\rho_i(\cdot)$ *is symmetric and* $\rho_i'(\mathbf{w}_i) \geq 0$ *for all* $\mathbf{w}_i \geq 0$. $(\star)$
2. *Each penalty function* $\rho_i(\cdot)$ *is applied element-wise on* $\mathbf{w}_i$*, which is of the form* $\mathbf{w}_i = \mathbf{K}_i\mathbf{y} + \mathbf{g}_i(\mathbf{x})$*, with filter matrix* $\mathbf{K}_i$ *and* $\mathbf{g}_i$ *not depending on* $\mathbf{y}$. $(\star\star)$

Proof. We first note that $\rho_i'(\cdot)$ is point symmetric since we assume $\rho_i(\cdot)$ to be symmetric in $(\star)$. Since $\rho_i'(\mathbf{w}_i) \geq 0$ for all $\mathbf{w}_i \geq 0$ we can rewrite $\rho_i'(\mathbf{w}_i)$ as

$$\rho_i'(\mathbf{w}_i) \equiv \tilde{\rho}_i(\mathbf{w}_i) \cdot \mathbf{w}_i \quad \text{with a} \quad \tilde{\rho}_i(\mathbf{w}_i) \geq 0. \tag{B.34}$$

Next, the gradient of an energy term as described in $(\star\star)$ is given as

$$\nabla_\mathbf{y} \rho_i(\mathbf{w}_i) = \mathbf{K}_i^\mathrm{T} \cdot \mathbf{C}_i \cdot (\mathbf{K}_i \cdot \mathbf{y} + \mathbf{g}_i(\mathbf{x})), \tag{B.35}$$

$$\text{with} \quad \mathbf{C}_i = \mathbf{D}\big(\tilde{\rho}_i\left(\mathbf{K}_i \cdot \mathbf{y} + \mathbf{g}_i(\mathbf{x})\right)\big). \tag{B.36}$$

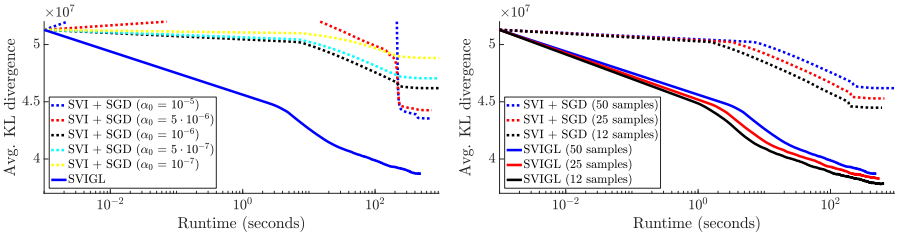
This yields a linearization by setting

$$\mathbf{A}_\mathbf{y}^i = \mathbf{K}_i^\mathrm{T} \cdot \mathbf{C}_i \cdot \mathbf{K}_i, \qquad \mathbf{b}_\mathbf{y}^i = \mathbf{K}_i^\mathrm{T} \cdot \mathbf{C}_i \cdot \mathbf{g}_i(\mathbf{x}). \tag{B.37}$$

Because $\mathbf{C}_i$ is diagonal and contains only non-negative elements (Eq. B.34), $\mathbf{A}_\mathbf{y}^i$ is positive semi-definite as

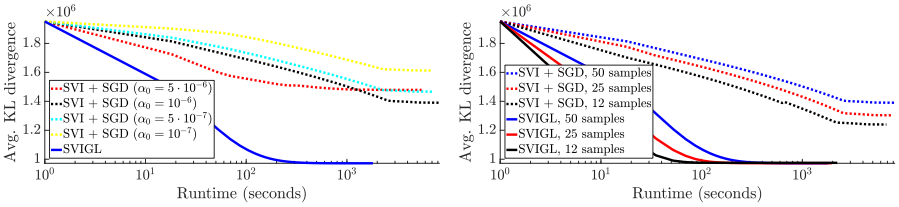
$$\mathbf{y}^\mathrm{T}\mathbf{A}_\mathbf{y}^i\mathbf{y} = \mathbf{y}^\mathrm{T}\mathbf{K}_i^\mathrm{T}\mathbf{C}_i\mathbf{K}_i\mathbf{y} = \mathbf{v}^\mathrm{T}\mathbf{C}_i\mathbf{v} \geq 0. \tag{B.38}$$

As positive semi-definite matrices are closed under summation, a matrix $\mathbf{A}_\mathbf{y}$ obtained of energy terms that fulfill $(\star)$ and $(\star\star)$ is positive semi-definite. □



(a) (b)

Figure B.1. Convergence of SVIGL and SVI with SGD for VI on the optical flow energy. In (a) we vary the step size of SGD. In (b) we vary the number of samples and iterations for SVIGL and SGD. We show average values of the validation dataset.



(a) (b)

Figure B.2. Unnormalized KL divergence *vs.* runtime for Poisson-Gaussian denoising with SVIGL and SVI with SGD with different step sizes (a) and with different numbers of samples and iterations (b). Values averaged over the BSDS test set.

B.4 HYPERPARAMETERS FOR SGD

Here, we show how we tuned hyperparameters for SVI with SGD. In all experiments we cut an initial step size α_0 by a factor of ten after each third of iterations. We first determine a good the initial step size α_0. Figure B.1a plots the unnormalized KL divergence *vs.* runtime for optical flow. For step sizes larger than 10^{-6} the KL divergence tends to deteriorate. For smaller step sizes we observe slow converge and hence choose $\alpha_0 = 10^{-6}$.

Following the same procedure, we perform several experiments for Poisson-Gaussian denoising and evaluate different settings for the initial step size parameter α_0 of SGD in Fig. B.2a. Again, an initial step size $\alpha_0 = 10^{-6}$ proves to be most effective. Smaller step sizes converge too slowly, while SGD with bigger step size values converges faster but to a worse local optimum. For an initial step size of $\alpha_0 = 10^{-5}$ optimization diverges immediately.

For both applications we observe faster convergence with a smaller sample size, but a larger number of iterations, *cf.* Figs. B.1b and B.2b. Hence, we use $|\mathcal{Z}| = 12$ and 4000 iterations of SGD for the experiments in Chapter 4.

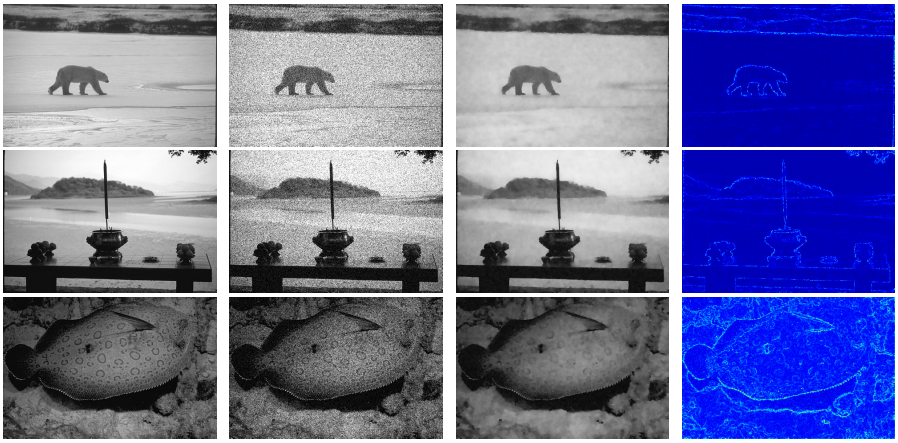


Figure B.3. Examples of ground truth (left), noisy images (second column), estimated clean images (third column), and uncertainty estimates (right) from SVIGL on the BSDS test set.

B.5 COMPARISON WITH PROBFLOWFIELDS

For the comparison with ProbFlowFields Table 4.2 we use the same EpicFlow (Revaud et al., 2015) energy with GSM potentials, keeping explicit indicator variables for mixture components, and using the same parameters as in (Wannenwetsch et al., 2017). Since SVIGL requires continuous distributions, we interleave SVIGL updates for the continuous flow variables with closed-form updates of the discrete indicator variables, where we replace analytical expectation values over the flow variables with Monte-Carlo approximations (*cf.* Eq. 4.11). This allows to update each indicator variable independently. Following Wannenwetsch et al. (2017), we use Bayesian optimization (Snoek et al., 2012) of the F1 score to tune weighting parameters λ_D and λ_S on a training set.

B.6 RESULTS OF POISSON-GAUSSIAN DENOISING

Fig. B.3 shows some example results of SVIGL applied to Poisson-Gaussian denoising on the BSDS dataset. High uncertainties can be observed especially on object boundaries. Due to the high amount of noise, a strong smoothness term maximizes the PSNR on the training set. Therefore, the denoised images tend to be rather smooth in general.

B.7 3D SURFACE RECONSTRUCTION

We now give more details on the application of SVIGL to 3D surface reconstruction. First, we restate the energy of Lipman et al. (2007), which is given by

$$E(Y,P,C) = \sum_{i=1}^{|Y|}\sum_{j=1}^{|P|}\|y_i - p_j\| \cdot h\big(\|c_i - p_j\|\big) - \sum_{i=1}^{|Y|}\sum_{i'=1}^{|C|}\lambda_i\|y_i - c_{i'}\| \cdot h\big(\|c_i - c_{i'}\|\big). \tag{B.39}$$

Here, $p_j \in P$ denote the noisy input points, $c_i \in C$ are the current estimates of the smoothed points, and $y_i \in Y$ the new estimates of the smoothed points. While the first part of the energy forces the new estimates to be close to the input points, the second term pushes the reconstructed points apart by penalizing points in Y that are too close to points in C. The contribution of each term is weighted by the Gaussian kernel $h(\cdot)$.

A closed-form solution to minimizing the above energy is given in (Lipman et al., 2007). This solution is then used in a fixed point scheme as

$$Y_{t+1} = \arg\min_{Y} E(Y,P,Y_t), \tag{B.40}$$

where Y_0 is initialized as a L_2 projection of the input points.

In a variational inference setting, closed-form updates are no longer possible due to introducing the additional variance variables σ of the variational posterior. Hence, we employ SVIGL updates instead. To be able to apply SVIGL, we require a linearization of the energy gradient. The specific form of the energy in Eq. (4.37) allows for a diagonal linearization:

$$\nabla_{y_i}E(Y,P,C) = \sum_{j\in J}(y_i - p_j)\frac{h(\|c_i - p_j\|)}{\|y_i - p_j\|} - \sum_{i'\in I}(x_i - c_{i'})\frac{h(\|c_i - c_{i'}\|)}{\|y_i - c_{i'}\|}. \tag{B.41}$$

In total, we run 10 iterations of Eq. (B.40). In each iteration, we compute a single SVIGL update with a sample set size of $|\mathcal{Z}| = 5$.

C

SUPPLEMENTAL MATERIAL FOR CHAPTER 5

In this supplemental material we give more details on the training protocol for single image super-resolution (SISR) and on the architectures for SISR and Gaussian denoising. Furthermore, we show extended quantitative and visual results for SISR.

C.1 ARCHITECTURES AND TRAINING DETAILS

A detailed summary of the used architectures can be found in the following tables:[1]

- Tables C.1 and C.2 show the architecture of embedding network and the temperature network within an N^3 block, respectively.
- Table C.3 shows the architecture of a DnCNN block used as local processing network in our N^3Net for denoising. The architecture of the whole N^3Net can be found in Table C.4.
- Table C.5 shows the architecture of a VDSR block used as local processing network in our N^3Net for single image super-resolution. The architecture of the whole N^3Net can be found in Table C.6.

Analogously to image denoising, the N^3 blocks for super-resolution extract 10×10 patches with a stride of 5 and patches are matched to other patches in a 80×80 region.

TRAINING DETAILS FOR SUPER-RESOLUTION. We follow the training protocol of Kim et al. (2016). Our training set consists of 291 images: The 200 images of the BSD500 training set and 91 images from Yang et al. (2010). In each of the 80 training epochs, we randomly crop 3833 patches of size 80×80 from each image and apply data augmentation by flipping and using a rotation $\in \{0°, 90°, 180°, 270°\}$. Our batchsize is 32. As in (Kim et al., 2016), we use the SGD optimizer with momentum of 0.9 and a weight decay of 10^{-4}. The initial learning rate is set to 0.1 and decayed by a factor of 10 every 20 epochs. Like Kim et al. (2016), we apply gradient clipping to stabilize training.

1 "K.", "S.", "P.", and "Feat." refer to the kernel size, stride, padding and number of feature channels, respectively.

Table C.1. Architecture of the embedding block.

Type	K., S., P.	Feat.
Input		8
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv	$3 \times 3, 1, 1$	8

Table C.2. Architecture of the block for predicting the temperature parameter.

Type	K., S., P.	Feat.
Input		8
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv	$3 \times 3, 1, 1$	1

Table C.3. Architecture of the 6 layer DnCNN blocks used for N^3Net for image denoising.

Type	K., S., P.	Feat.
Input		1 if first block 64 else
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv	$3 \times 3, 1, 1$	1 if last block 8 else
Skip		

Table C.4. Architecture of N^3Net for image denoising.

Type	k	Feat.
Input		1
DnCNN block		8
N^3 block	7	64
DnCNN block		8
N^3 block	7	64
DnCNN block		1

C.2 EXTENDED ABLATION STUDY FOR GAUSSIAN DENOISING

We conduct further ablation studies on the task of removing additive white Gaussian noise, extending the results of Section 5.6.1. We basically want to discern the effect of adding a *single* KNN or N^3 block, respectively, and the effect of training the baseline model on bigger patch sizes. Table C.7 shows these results. We make the following observations: First, for $d = 6$ our N^3 block outperforms simple stacking of DnCNN networks as well as using a KNN block by a significant margin, for both $\sigma = 25$ and 70. Second, for $d = 17$ stacking two full networks performs poorly as training becomes more difficult with the increased depth. Interestingly, N^3 can remedy some of the ill effects. Third, increasing the receptive field for the baseline DnCNN using more layers does not always help (*cf.* $2 \times$ DnCNN, $d = 17$ in Table C.7). This is in contrast to our approach that allows increasing the receptive field without having many layers or parameters. Fourth, training on larger patch sizes does not benefit the baseline DnCNN model, *cf.* baseline *(i)* in Table 5.1.

Table C.5. Architecture of the 7 layer VDSR blocks used for N^{3}Net for super resolution.

Type	K., S., P.	Feat.
Input		1 if first block 64 else
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv/BN/ReLU	$3 \times 3, 1, 1$	64
Conv	$3 \times 3, 1, 1$	1 if last block 8 else
Skip		

Table C.6. Architecture of N^{3}Net for super resolution.

Type	k	Feat.
Input		1
VDSR block		8
N^3 block	7	64
VDSR block		8
N^3 block	7	64
VDSR block		1

Table C.7. PSNR (dB) on Urban100 for different architectures on gray-scale image denoising. Models are trained on 80×80 patches.

	d=6		d=17	
Model	σ=25	σ=70	σ=25	σ=70
$1 \times$ DnCNN	29.04	23.39	29.74	24.36
$2 \times$ DnCNN	29.59	24.19	29.48	13.77
$2 \times$ DnCNN, KNN block (k=7)	29.82	24.63	**29.85**	22.49
$2 \times$ DnCNN, N^3 block (k=7)	**29.99**	**24.91**	29.82	**24.18**

C.3 SUPER-RESOLUTION RESULTS

Table C.9 and Table C.8 show results for single image super-resolution on two further datasets: The full BSD500 validation set consisting of 100 images (BSD100), and Urban100. We observe a consistent gain of N^{3}Net compared to the very strong baseline VDSR on both datasets and all super-resolution factors. Moreover, the performance of the other non-local methods falls short compared to both the baseline and our N^{3}Net. Figure C.1 shows visual results for our method and VDSR. We can see that N^{3}Net produces sharper details than VDSR, leading to perceptually more pleasing images despite the PSNR values being relatively close.

Table C.8. PSNR (dB) values for single image super-resolution on Urban100.

	Bicubic	SelfEx	WSD-SR	MemNet	MDSR	VDSR	N^{3}Net
$\times 2$	26.88	29.54	30.29	31.31	32.84	30.76	30.80
$\times 3$	24.46	26.44	26.95	27.56	28.79	27.14	27.19
$\times 4$	23.14	24.79	25.16	25.50	26.67	25.18	25.23

Table C.9. PSNR (dB) values for single image super-resolution on BSD100. WSD-SR does not provide results for BSD100.

	Bicubic	SelfEx	MemNet	MDSR	VDSR	N^3Net
×2	29.56	31.18	32.05	32.29	31.90	31.98
×3	27.21	28.29	28.95	29.25	28.82	28.91
×4	25.96	26.84	27.38	27.72	27.29	27.34

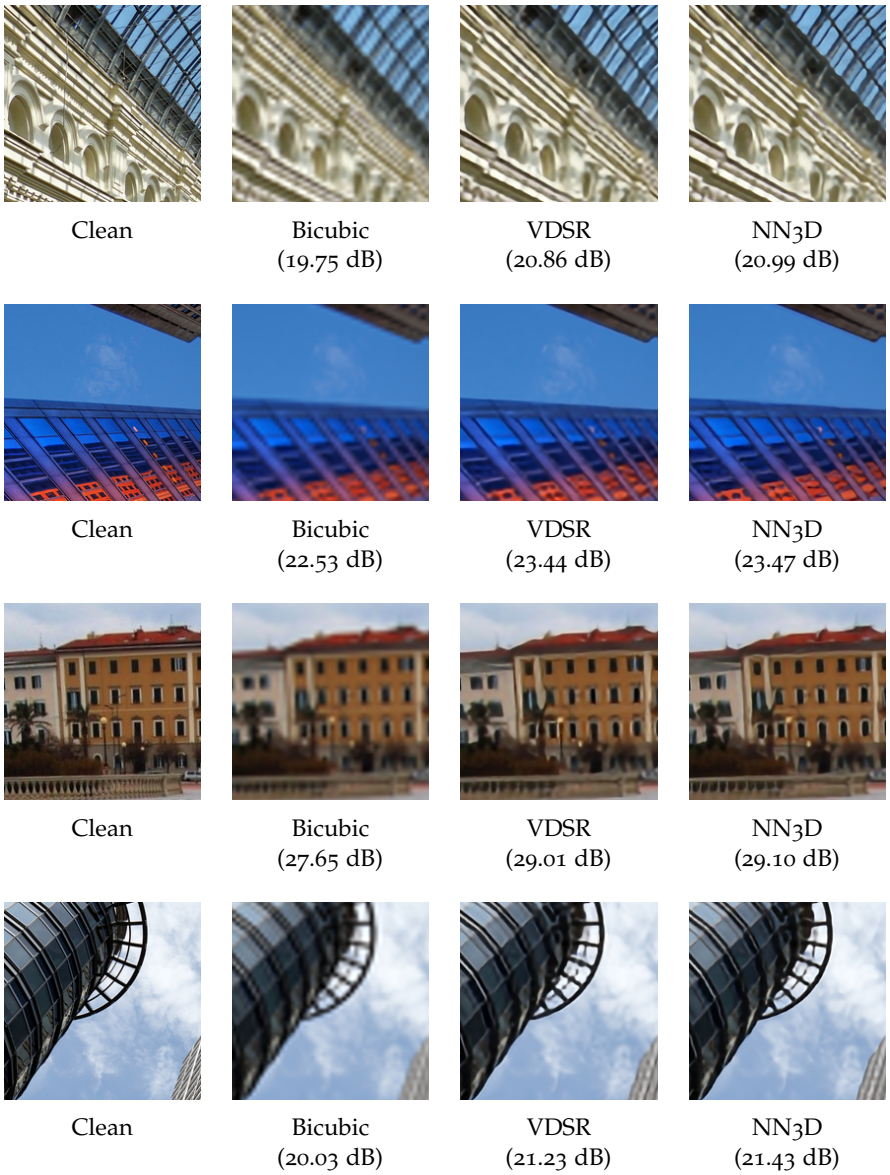


Figure C.1. Super-resolution results (cropped for better display) and PSNR values on four images from Urban100 with a super-resolution factor of 4.

BIBLIOGRAPHY

Abdelhamed, Abdelrahman, Stephen Lin, and Michael S. Brown (June 2018). "A high-quality denoising dataset for smartphone cameras." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 1692–1700.

Abdelhamed, Abdelrahman, Radu Timofte, and Michael S. Brown (June 2019a). "NTIRE 2019 challenge on real image denoising: Methods and results." In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. Long Beach, Carlifornia.

Abdelhamed, Abdelrahman, Marcus A. Brubaker, and Michael S. Brown (Oct. 2019b). "Noise Flow: Noise Modeling with Conditional Normalizing Flows." In: *Proceedings of the Seventeenth IEEE International Conference on Computer Vision*. Seoul, South Korea, pp. 3165–3173.

Agustsson, Eirikur and Radu Timofte (July 2017). "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Trends in Image Restoration and Enhancement Workshop (NTIRE)*. Honolulu, Hawaii, pp. 126–135.

Aharon, Michal, Michael Elad, and Alfred Bruckstein (Nov. 2006). "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation." In: *IEEE Transactions on Image Processing* 54.11, pp. 4311–4322.

Ahn, Byeongyong and Nam Ik Cho (2017). "Block-matching convolutional neural network for image denoising." In: *arXiv:1704.00524 [cs.CV]*.

Alain, Guillaume and Yoshua Bengio (Jan. 2014). "What regularized auto-encoders learn from the data-generating distribution." In: *Journal of Machine Learning Research* 15.1, pp. 3563–3593.

Anaya, Josue and Adrian Barbu (Feb. 2018). "RENOIR - A dataset for real low-light image noise reduction." In: *Journal of Visual Communication and Image Representation* 51, pp. 144–154.

Andrews, D. F. and C. L. Mallows (1974). "Scale Mixtures of Normal Distributions." In: *Journal of the Royal Statistical Society. Series B, Statistical Methodology* 36.1, pp. 99–102.

Arjovsky, Martin, Soumith Chintala, and Léon Bottou (Aug. 2017). "Wasserstein Generative Adversarial Networks." In: *Proceedings of the 34rd International Conference on Machine Learning*. Vol. 70. Sydney, Australia, pp. 214–223.

Azzari, Lucio and Alessandro Foi (May 2014). "Gaussian-Cauchy mixture modeling for robust signal-dependent noise estimation." In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Florence, Italy, pp. 5357–5361.

Bae, Woong, Jae Jun Yoo, and Jong Chul Ye (July 2017). "Beyond Deep Residual Learning for Image Restoration: Persistent Homology-Guided Manifold Simplification." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Trends in Image Restoration and Enhancement Workshop (NTIRE)*. Honolulu, Hawaii, pp. 145–153.

Bailer, Christian, Bertram Taetz, and Didier Stricker (Dec. 2015). "Flow Fields: Dense Correspondence Fields for Highly Accurate Large Displacement Optical Flow Estimation." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile: IEEE, pp. 2030–2038.

Barron, Jonathan T. (June 2019). "A More General Robust Loss Function." In: pp. 4331–4339.

Barron, Jonathan T. and Ben Poole (2016). "The Fast Bilateral Solver." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9907. Lecture Notes in Computer Science. Springer, pp. 617–632.

Benfold, Ben and Ian Reid (June 2011). "Stable Multi-Target Tracking in Real-Time Surveillance Video." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado Springs, Colorado.

Bennett, William Ralph (1948). "Spectra of quantized signals." In: *Bell System Technical Journal* 27.3, pp. 446–472.

Bevilacqua, Marco, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel (Sept. 2012). "Low-complexity single-image super-resolution based on nonnegative neighbor embedding." In: *Proceedings of the British Machine Vision Conference*. Surrey, UK, pp. 135.1–135.10.

Bigdeli, Siavash Arjomand, Matthias Zwicker, Paolo Favaro, and Meiguang Jin (2017). "Deep Mean-Shift Priors for Image Restora-

tion." In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp. 763–772.

Bioucas-Dias, José and Gonçalo Valadao (Mar. 2007). "Phase Unwrapping Via Graph Cuts." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.3, pp. 698–709.

Bioucas-Dias, Jose, Vladimir Katkovnik, Jaakko Astola, and Karen Egiazarian (2008). "Absolute phase estimation: adaptive local denoising and global unwrapping." In: *OSA Applied Optics* 47.29, pp. 5358–5369.

Bischof, Christian H., Ali Bouaricha, Peyvand M. Khademi, and Jorge J. Moré (May 1997). "Computing Gradients in Large-Scale Optimization Using Automatic Differentiation." In: *INFORMS Journal on Computing* 9.2, pp. 185–194.

Black, Michael J. and P. Anandan (June 1991). "Robust Dynamic Motion Estimation over Time." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Lahaina, Maui, Hawaii, pp. 296–302.

Black, Michael J., Guillermo Sapiro, David H. Marimont, and David Heeger (Mar. 1998). "Robust Anisotropic Diffusion." In: *IEEE Transactions on Image Processing* 7.3, pp. 421–432.

Blake, Andrew, Pushmeet Kohli, and Carsten Rother, eds. (2011). *Markov Random Fields for Vision and Image Processing*. MIT Press.

Blau, Yochai and Tomer Michaeli (June 2018). "The perception-distortion tradeoff." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 6228–6237.

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2003). "Latent Dirichlet allocation." In: *Journal of Machine Learning Research* 3.1, pp. 993–1022.

Blu, Thierry and Florian Luisier (2007). "The SURE-LET approach to image denoising." In: *IEEE Transactions on Image Processing* 16.11, pp. 2778–2786.

Böhm, M. et al. (Jan. 1998). "High Dynamic Range Image Sensors in Thin Film on ASIC Technology for Automotive Applications." In: *Advanced Microsystems for Automotive Applications*. Ed. by Detlef Egbert Ricken and Wolfgang Gessner. Springer, pp. 157–172.

Boyd, Stephen, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. (2011). "Distributed optimization and statistical learning via the alternating direction method of multipliers." In: *Foundations and Trends in Machine learning* 3.1, pp. 1–122.

Boykov, Yuri, Olga Veksler, and Ramin Zabih (Nov. 2001). "Fast Approximate Energy Minimization via Graph Cuts." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.11, pp. 1222–1239.

Brooks, Tim, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron (June 2019). "Unprocessing Images for Learned Raw Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Long Beach, Carlifornia, pp. 11036–11045.

Brox, Thomas, Andrés Bruhn, Nils Papenberg, and Joachim Weickert (2004). "High accuracy optical flow estimation based on a theory for warping." In: *Proceedings of the Eighth European Conference on Computer Vision*. Vol. 3024. Lecture Notes in Computer Science. Springer, pp. 25–36.

Buades, A[ntoni], B[artomeu] Coll, and J[ean]-M[ichel] Morel (2004). "A Review of Image Denoising Algorithms, with a New One." In: *SIAM Multiscale Modeling and Simulation* 4.2, pp. 490–530.

Buades, Antoni, Bartomeu Coll, and Jean-Michel Morel (June 2005a). "A Non-Local Algorithm for Image Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego, California, pp. 60–65.

Buades, Antoni, Bartomeu Coll, and Jean-Michel Morel (Mar. 2005b). "Image Denoising by Non-Local Averaging." In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Vol. 2. Philadelphia, Pennsylvania, pp. 25–28.

Burger, Harold C., Christian J. Schuler, and Stefan Harmeling (June 2012). "Image denoising: Can plain Neural Networks compete with BM3D?" In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Providence, Rhode Island, pp. 2392–2399.

Butler, Daniel J., Jonas Wulff, Garrett B. Stanley, and Micheal J. Black (2012). "A naturalistic open source movie for optical flow evaluation." In: *Proceedings of the 12th European Conference on Computer Vision*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Lecture Notes in Computer Science. Springer, pp. 611–625.

Byrd, Richard H., Peihuang Lu, Jorge Nocedal, and Ciyou Zhu (1995). "A Limited Memory Algorithm for Bound Constrained Optimization." In: *SIAM Journal on Scientific Computing* 16.5, pp. 1190–1208.

Chambolle, Antonin and Thomas Pock (2011). "A first-order primal-dual algorithm for convex problems with applications to imaging." In: *Journal of mathematical imaging and vision* 40.1, pp. 120–145.

Chambolle, Antonin, Ronald A. DeVore, Nam-Yong Lee, and Bradley J Lucier (1998). "Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage." In: *IEEE Transactions on Image Processing* 7.3, pp. 319–335.

Chantas, Giannis, Nikolaos Galatsanos, Aristidis Likas, and Michael Saunders (Oct. 2008). "Variational Bayesian Image Restoration Based on a Product of t-Distributions Image Prior." In: *IEEE Transactions on Image Processing* 17.10, pp. 1795–1805.

Chatterjee, Priyam and Peyman Milanfar (2010). "Is denoising dead?" In: *IEEE Transactions on Image Processing* 19.4, pp. 895–911.

Chatterjee, Priyam and Peyman Milanfar (Apr. 2012). "Patch-Based Near-Optimal Image Denoising." In: *IEEE Transactions on Image Processing* 21.4, pp. 1635–1649.

Chavez, Sofia, Qing-San Xiang, and Li An (Aug. 2002). "Understanding phase maps in MRI: A new cutline phase unwrapping method." In: *IEEE Transactions on Medical Imaging* 21.8, pp. 966–977.

Chen, Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun (June 2018a). "Learning to see in the dark." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 3291–3300.

Chen, Fei, Lei Zhang, and Huimin Yu (Dec. 2015a). "External patch prior guided internal clustering for image denoising." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 603–611.

Chen, Guangyong, Fengyuan Zhu, and Pheng Ann Heng (Dec. 2015b). "An efficient statistical method for image noise level estimation." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 477–485.

Chen, Jia and Chi-Keung Tang (June 2007). "Spatio-temporal Markov random field for video denoising." In: *Proceedings of the IEEE Com-*

puter Society Conference on Computer Vision and Pattern Recognition. Minneapolis, Minnesota, pp. 2232–2239.

Chen, Jingwen, Jiawei Chen, Hongyang Chao, and Ming Yang (June 2018b). "Image Blind Denoising With Generative Adversarial Network Based Noise Modeling." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 3155–3164.

Chen, Yu, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang (June 2018c). "FSRNet: End-to-End Learning Face Super-Resolution with Facial Priors." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 2492–2501.

Chen, Yunjin and Thomas Pock (2017). "Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.6, pp. 1256–1272.

Chen, Yunjin, Wei Yu, and Thomas Pock (June 2015c). "On learning optimized reaction diffusion processes for effective image restoration." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 5261–5269.

Coleman, Paul H. and Luciano Pietronero (1992). "The fractal structure of the universe." In: *Elsevier Physics Reports* 213.6, pp. 311–389.

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Scharwächter, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele (June 2015). *The Cityscapes Dataset*. CVPR 2015 Workshop on The Future of Datasets in Computer Vision. Abstract. Boston, Massachusetts.

Courbariaux, Matthieu, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio (2016). "Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1." In: *arXiv:1602.02830 [cs.LG]*.

Cruz, Cristóvão, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian (2018a). "Nonlocality-Reinforced Convolutional Neural Networks for Image Denoising." In: *IEEE Signal Processing Letters* 25.8, pp. 1216–1220.

Cruz, Cristóvão, Rakesh Mehta, Vladimir Katkovnik, and Karen O. Egiazarian (Mar. 2018b). "Single Image Super-Resolution Based on

Wiener Filter in Similarity Domain." In: *IEEE Transactions on Image Processing* 27.2, pp. 1376–1389.

Dabov, Kostadin, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian (2006). "Image denoising with block-matching and 3D filtering." In: *Electronic Imaging '06, Proc. SPIE 6064, No. 6064A-30*.

Dabov, Kostadin, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian (Aug. 2007). "Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering." In: *IEEE Transactions on Image Processing* 16.8, pp. 2080–2095.

Debevec, Paul E. and Jitendra Malik (Aug. 1997). "Recovering High Dynamic Range Radiance Maps from Photographs." In: *Computer Graphics (Proceedings of ACM SIGGRAPH)*. Los Angeles, CA, USA, pp. 369–378.

Deledalle, Charles-Alban, Loïc Denis, and Florence Tupin (2012). "How to compare noisy patches? Patch similarity beyond Gaussian noise." In: *International Journal of Computer Vision* 99.1, pp. 86–102.

Deng, Xin (2018). "Enhancing image quality via style transfer for single image super-resolution." In: *IEEE Signal Processing Letters* 25.4, pp. 571–575.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2017). "Density estimation using Real NVP." In:

Dong, Weisheng, Lei Zhang, Guangming Shi, and Xin Li (Apr. 2013). "Nonlocally Centralized Sparse Representation for Image Restoration." In: *IEEE Transactions on Image Processing* 22.4, pp. 1620–1630.

Donoho, David L. (May 1995). "Denoising by Soft-Thresholding." In: *IEEE Transactions on Information Theory* 41.3, pp. 613–627.

Drago, Frédéric, Karol Myszkowski, Thomas Annen, and Norishige Chiba (2003). "Adaptive logarithmic mapping for displaying high contrast scenes." In: 22.3, pp. 419–426.

Duchi, John, Elad Hazan, and Yoram Singer (2011). "Adaptive subgradient methods for online learning and stochastic optimization." In: *Journal of Machine Learning Research* 12.7, pp. 2121–2159.

Dwork, Cynthia, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth (June 2015a). "Preserving statistical validity in adaptive data analysis." In: *Proceedings of the Forty-Seventh*

Annual ACM Symposium on Theory of Computing (STOC). ACM. Portland, Oregon, USA, pp. 117–126.

Dwork, Cynthia, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth (2015b). "The reusable holdout: Preserving validity in adaptive data analysis." In: *Science* 349.6248, pp. 636–638.

El Gamal, Abbas and Helmy Eltoukhy (May 2005). "CMOS Image Sensors." In: *IEEE Circuits and Devices Magazine* 21.3, pp. 6–20.

Elad, Michael and Michal Aharon (June 2006). "Image Denoising Via Learned Dictionaries and Sparse Representations." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 1. New York, New York, pp. 895–900.

European Machine Vision Association (2012). *EMVA Standard 1288: Standard for characterization of Image Sensors and Cameras*. URL: http://www.emva.org/wp-content/uploads/EMVA1288-3.1rc1.pdf.

Everingham, Mark, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman (2012). *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*. URL: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fan, Kai, Ziteng Wang, Jeffrey M. Beck, James T. Kwok, and Katherine A. Heller (2015). "Fast Second-Order Stochastic Backpropagation for Variational Inference." In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Vol. 28, pp. 1387–1395.

Field, David J. (Dec. 1987). "Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells." In: *Journal of the Optical Society of America. Series A, Optics and Image Science* 4.12, pp. 2379–2394.

Field, David J. (1993). "Scale-invariance and self-similar 'wavelet'transforms: An analysis of natural scenes and mammalian visual systems." In: *Wavelets, fractals, and Fourier transforms: New developments and new applications*, pp. 151–193.

Foi, Alessandro (Dec. 2009). "Clipped noisy images: Heteroskedastic modeling and practical denoising." In: *Elsevier Signal Processing* 89.12, pp. 2609–2629.

Foi, Alessandro, Sakari Alenius, Vladimir Katkovnik, and Karen Egiazarian (Oct. 2007). "Noise Measurement for Raw-Data of Dig-

ital Imaging Sensors by Automatic Segmentation of Nonuniform Targets." In: *IEEE Sensors Journal* 7.10, pp. 1456–1461.

Foi, Alessandro, Mejdi Trimeche, Vladimir Katkovnik, and Karen Egiazarian (Oct. 2008). "Practical Poissonian-Gaussian Noise Modeling and Fitting for Single-Image Raw-Data." In: *IEEE Transactions on Image Processing* 17.10, pp. 1737–1754.

Frosio, Iuri and Jan Kautz (2019). "Statistical Nearest Neighbors for Image Denoising." In: *IEEE Transactions on Image Processing* 28.2, pp. 723–738.

Fuji Photo Film Co., Ltd. (2003). *4th-Generation Super CCD*. URL: http://www.fujifilmusa.com/shared/bin/4thGenSUPERCCDBrochure.pdf.

Gabay, Daniel and Bertrand Mercier (1976). "A dual algorithm for the solution of nonlinear variational problems via finite element approximation." In: *Elsevier Computers & Mathematics with Applications* 2.1, pp. 17–40.

Gao, Qi and Stefan Roth (2012). "How well do filter-based MRFs model natural images?" In: *Pattern Recognition, Proceedings of the 34th DAGM-Symposium*. Ed. by A. Pinz, T. Pock, H. Bischof, and F. Leberl. Vol. 7476. Lecture Notes in Computer Science. Springer, pp. 62–72.

Geman, Donald and George Reynolds (Mar. 1992). "Constrained Restoration and the Recovery of Discontinuities." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.3, pp. 367–383.

Geman, Stuart and Donald Geman (Nov. 1984). "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6, pp. 721–741.

Glorot, Xavier, Antoine Bordes, and Yoshua Bengio (Apr. 2011). "Deep sparse rectifier neural networks." In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. Ft. Lauderdale, Florida: PMLR, pp. 315–323.

Glowinski, Roland and A Marroco (1975). "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires." In: *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9.R2, pp. 41–76.

Goldberger, Jacob, Geoffrey E. Hinton, Sam T. Roweis, and Ruslan R. Salakhutdinov (2006). "Neighbourhood components analysis." In: *Advances in Neural Information Processing Systems*. Ed. by Y. Weiss, B. Schölkopf, and J. Platt, pp. 513–520.

Goldstein, Richard M., Howard A. Zebker, and Charles L. Werner (July 1988). "Satellite radar interferometry: Two-dimensional phase unwrapping." In: *IEEE Radio Science* 23.4, pp. 713–720.

Gomez, Aidan N., Mengye Ren, Raquel Urtasun, and Roger B. Grosse (2017). "The reversible residual network: Backpropagation without storing activations." In: *Advances in Neural Information Processing Systems*, pp. 2214–2224.

Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio (2014). "Generative adversarial nets." In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, pp. 2672–2680.

Gorthi, Sai Siva and Pramod Rastogi (2010). "Fringe projection techniques: whither we are?" In: *Elsevier Optics and Lasers in Engineering* 48.2, pp. 133–140.

Granados, Miguel, Boris Ajdin, Michael Wand, Christian Theobalt, Hans-Peter Seidel, and Hendrik P. A. Lensch (June 2010). "Optimal HDR reconstruction with linear digital cameras." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, California, pp. 215–222.

Graves, Alex, Greg Wayne, and Ivo Danihelka (2014). "Neural Turing machines." In: *arXiv:1410.5401 [cs.NE]*.

Grossberg, Michael D. and Shree K. Nayar (Oct. 2003). "High Dynamic Range from Multiple Images: Which Exposures to Combine?" In: *IEEE International Conference on Computer Vision, Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*. Nice, France.

Gu, Shuhang, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng (June 2014). "Weighted Nuclear Norm Minimization with Application to Image Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 2862–2869.

Guizar-Sicairos, Manuel, Samuel T. Thurman, and James R. Fienup (Jan. 2008). "Efficient subpixel image registration algorithms." In: *Optics Letters* 33.2, pp. 156–158.

Guo, Shi, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang (June 2019). "Toward Convolutional Blind Denoising of Real Photographs." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Long Beach, Carlifornia, pp. 1712–1722.

Haber, Eldad and Lars Ruthotto (2017). "Stable architectures for deep neural networks." In: *Inverse Problems* 34.1.

Hahnloser, Richard L. T. (1998). "On the piecewise analysis of networks of linear threshold neurons." In: *Neural Networks* 11.4, pp. 691–697.

He, Kaiming, Jian Sun, and Xiaoou Tang (2013). "Guided image filtering." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.6, pp. 1397–1409.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick (Oct. 2017). "Mask R-CNN." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. IEEE. Venice, Italy, pp. 2980–2988.

Healy, Glenn E. and Raghava Kondepudy (Mar. 1994). "Radiometric CCD camera calibration and noise estimation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.3, pp. 267–276.

Heide, Felix et al. (2014). "FlexISP: A flexible camera image processing framework." In: *ACM Transactions on Graphics* 33.6, p. 231.

Hoffman, Matthew D, David M. Blei, Chong Wang, and John William Paisley (2013). "Stochastic variational inference." In: *Journal of Machine Learning Research* 14.1, pp. 1303–1347.

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam (2017). "Mobilenets: Efficient convolutional neural networks for mobile vision applications." In: *arXiv:1704.04861 [cs.CV]*.

Huang, Jia-Bin, Abhishek Singh, and Narendra Ahuja (June 2015). "Single image super-resolution from transformed self-exemplars." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 5197–5206.

Idier, Jérôme (2001). "Convex half-quadratic criteria and interacting auxiliary variables for image restoration." In: *IEEE Transactions on Image Processing* 10.7, pp. 1001–1009.

Ignatov, Andrey, Radu Timofte, Przemyslaw Szczepaniak, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool (2018a). "AI benchmark: Running deep neural networks on android smartphones." In: Lecture Notes in Computer Science 11133. Ed. by L. Leal-Taixé and S. Roth, pp. 288–314.

Ignatov, Andrey, Radu Timofte, Thang Van Vu, Tung Minh Luu, Trung X. Pham, Cao Van Nguyen, Yongwoo Kim, Jae-Seok Choi, Munchurl Kim, Jie Huang, et al. (2018b). "PIRM challenge on perceptual image enhancement on smartphones: Report." In: Lecture Notes in Computer Science 11133. Ed. by L. Leal-Taixé and S. Roth, pp. 315–333.

Im, Daniel Jiwoong, Sungjin Ahn, Roland Memisevic, and Yoshua Bengio (Feb. 2017). "Denoising Criterion for Variational Auto-Encoding Framework." In: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. San Francisco, CA, USA: AAAI, pp. 2059–2065.

Ioffe, Sergey and Christian Szegedy (July 2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." In: *Proceedings of the 32nd International Conference on Machine Learning*. Lille, France, pp. 448–456.

Jain, Viren and H. Sebastian Seung (2009). "Natural Image Denoising with Convolutional Networks." In: *Advances in Neural Information Processing Systems*. Ed. by D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou. Vol. 21, pp. 769–776.

Jancsary, Jeremy, Sebastian Nowozin, and Carsten Rother (2012). "Loss-Specific Training of Non-Parametric Image Restoration Models: A New State of the Art." In: *Proceedings of the 12th European Conference on Computer Vision*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Vol. 7578. Lecture Notes in Computer Science. Springer, pp. 112–125.

Jang, Eric, Shixiang Gu, and Ben Poole (2017). "Categorical reparameterization with Gumbel-softmax." In: *ICLR*.

Kamilov, Ulugbek S., Ioannis N. Papadopoulos, Morteza H. Shoreh, Demetri Psaltis, and Michael Unser (2015). "Isotropic inverse-problem approach for two-dimensional phase unwrapping." In: *Journal of the Optical Society of America A* 32.6, pp. 1092–1100.

Karaimer, Hakki Can and Michael S. Brown (2016). "A Software Platform for Manipulating the Camera Imaging Pipeline." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B.

Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9905. Lecture Notes in Computer Science. Springer, pp. 429–444.

Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee (June 2016). "Accurate Image Super-Resolution Using Very Deep Convolutional Networks." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 1646–1654.

Kingma, Diederik P. and Max Welling (2014). "Auto-Encoding Variational Bayes." In: *Proceedings of the International Conference on Learning Representations*, Banff, Canada.

Kingma, Diederik and Jimmy Ba (2015). "Adam: A method for stochastic optimization." In: *ICLR*.

Kingma, Durk P. and Prafulla Dhariwal (2018). "Glow: Generative flow with invertible 1x1 convolutions." In: *Advances in Neural Information Processing Systems*, pp. 10236–10245.

Krähenbühl, Philipp and Vladlen Koltun (2011). "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials." In: *Advances in Neural Information Processing Systems*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Vol. 24. Granada, Spain: NIPS Fnd., pp. 109–117.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). "Imagenet classification with deep convolutional neural networks." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25, pp. 1097–1105.

Lang, Florian, Tobias Plötz, and Stefan Roth (2017). "Robust Multi-Image HDR Reconstruction for the Modulo Camera." In: *Proceedings of the 39th German Conference on Pattern Recognition*. Ed. by Volker Roth and Thomas Vetter. Vol. 10496. Lecture Notes in Computer Science. Springer, pp. 78–89.

Lebrun, Marc, Antoni Buades, and Jean-Michel Morel (2013). "A nonlocal Bayesian image denoising algorithm." In: *SIAM Journal on Imaging Sciences* 6.3, pp. 1665–1688.

Lebrun, Marc, Miguel Colom, and Jean-Michel Morel (Oct. 2014). "The noise clinic: A universal blind denoising algorithm." In: *Proceedings of the IEEE International Conference on Image Processing*. Paris, France, pp. 2674–2678.

Ledig, Christian et al. (June 2018). "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 4681–4690.

Lefkimmiatis, Stamatios (July 2017). "Non-local color image denoising with convolutional neural networks." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 5882–5891.

Lefkimmiatis, Stamatios (June 2018). "Universal Denoising Networks: A Novel CNN-based Network Architecture for Image Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 3204–3213.

Levin, Anat and Boaz Nadler (June 2011). "Natural Image Denoising: Optimality and Inherent Bounds." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado Springs, Colorado.

Levin, Anat, Yair Weiss, Fredo Durand, and William T. Freeman (June 2011). "Efficient Marginal Likelihood Optimization in Blind Deconvolution." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado Springs, Colorado: IEEE, pp. 2657–2664.

Levin, Anat, Boaz Nadler, Fredo Durand, and William T. Freeman (2012). "Patch complexity, finite pixel correlations and optimal denoising." In: *Proceedings of the 12th European Conference on Computer Vision*. Ed. by A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid. Vol. 7576. Lecture Notes in Computer Science. Springer, pp. 73–86.

Lichtsteiner, Patrick, Christoph Posch, and Tobi Delbruck (2008). "A 128×128 120 dB 15μs Latency Asynchronous Temporal Contrast Vision Sensor." In: *IEEE Journal of Solid-state Circuits* 43.2, pp. 566–576.

Likas, Aristidis C. and Nikolas P. Galatsanos (Aug. 2004). "A variational approach for Bayesian blind image deconvolution." In: *IEEE Transactions on Signal Processing* 52.8, pp. 2222–2233.

Lim, Bee, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee (July 2017). "Enhanced deep residual networks for single image super-resolution." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Trends in Image Restora-*

tion and Enhancement Workshop (NTIRE). Honolulu, Hawaii, pp. 136–144.

Lipman, Yaron, Daniel Cohen-Or, David Levin, and Hillel Tal-Ezer (2007). "Parameterization-free projection for geometry reconstruction." In: 26.3, p. 22.

Liu, Ce, Richard Szeliski, Sing Bing Kang, C. Lawrence Zitnick, and William T. Freeman (Feb. 2008). "Automatic Estimation and Removal of Noise from a Single Image." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2, pp. 299–314.

Liu, Ding, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas Huang (2018). "Non-Local Recurrent Network for Image Restoration." In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett.

Liu, Pengju, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo (July 2017). "Multi-level Wavelet-CNN for Image Restoration." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Trends in Image Restoration and Enhancement Workshop (NTIRE)*. Honolulu, Hawaii, pp. 773–782.

Liu, Xinhao, Masayuki Tanaka, and Masatoshi Okutomi (Dec. 2013). "Single-Image Noise Level Estimation for Blind Denoising." In: *IEEE Transactions on Image Processing* 22.12, pp. 5226–5237.

Liu, Xinhao, Masayuki Tanaka, and Masatoshi Okutomi (2014). "Practical Signal-Dependent Noise Parameter Estimation From a Single Noisy Image." In: *IEEE Transactions on Image Processing* 23.10, pp. 4361–4371.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell (June 2015). "Fully convolutional networks for semantic segmentation." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts, pp. 3431–3440.

Loose, Markus, Karlheinz Meier, and Johannes Schemmel (Apr. 2001). "A self-calibrating single-chip CMOS camera with logarithmic response." In: *IEEE Journal of Solid-State Circuits* 36.4, pp. 586–596.

Lotan, Or and Michal Irani (June 2016). "Needle-match: Reliable patch matching under high uncertainty." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 439–448.

Lowe, David G. (Nov. 2004). "Distinctive Image Features from Scale-Invariant Keypoints." In: *International Journal of Computer Vision* 60.2, pp. 91–110.

Lu, Yiping, Aoxiao Zhong, Quanzheng Li, and Bin Dong (2018). "Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations." In: *Proceedings of the Sixth International Conference on Learning Representations*. Vancouver, Canada.

Lucas, Bruce D. and Takeo Kanade (Aug. 1981). "An Iterative Image Registration Technique with an Application to Stereo Vision." In: *Proceedings of the International Joint Conference on Artificial Intelligence*. Vancouver, British Columbia, pp. 674–679.

Luisier, Florian and Thierry Blu (2008). "SURE-LET multichannel image denoising: interscale orthonormal wavelet thresholding." In: *IEEE Transactions on Image Processing* 17.4, pp. 482–492.

Luisier, Florian, Thierry Blu, and Michael Unser (2011). "Image denoising in mixed Poisson-Gaussian noise." In: *IEEE Transactions on Image Processing* 20.3, pp. 696–708.

Ma, Chao, Chih-Yuan Yang, Xiaokang Yang, and Ming-Hsuan Yang (2017a). "Learning a no-reference quality metric for single-image super-resolution." In: *Computer Vision and Image Understanding* 158, pp. 1–16.

Ma, Kede, Zhengfang Duanmu, Qingbo Wu, Zhou Wang, Hongwei Yong, Hongliang Li, and Lei Zhang (Feb. 2017b). "Waterloo Exploration Database: New Challenges for Image Quality Assessment Models." In: *IEEE Transactions on Image Processing* 26.2, pp. 1004–1016.

Maddison, Chris J., Andriy Mnih, and Yee Whye Teh (2017). "The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables." In: *ICLR*.

Mairal, Julien, Michael Elad, and Guillermo Sapiro (Jan. 2008). "Sparse Representation for Color Image Restoration." In: *IEEE Transactions on Image Processing* 17.1, pp. 53–69.

Mairal, Julien, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman (Oct. 2009). "Non-local Sparse Models for Image Restoration." In: *Proceedings of the Twelfth IEEE International Conference on Computer Vision*. Kyoto, Japan, pp. 2272–2279.

Mäkitalo, Markku and Alessandro Foi (Jan. 2013). "Optimal Inversion of the Generalized Anscombe Transformation for Poisson-Gaussian Noise." In: *IEEE Transactions on Image Processing* 22.1, pp. 91–103.

Mäkitalo, Markku and Alessandro Foi (2014). "Noise Parameter Mismatch in Variance Stabilization, With an Application to Poisson-Gaussian Noise Estimation." In: *IEEE Transactions on Image Processing* 23.12, pp. 5348–5359.

Mäkitalo, Markku and Alessandro Foi (Oct. 2014). "Noise parameter mismatch in variance stabilization, with an application to Poisson–Gaussian noise estimation." In: *IEEE Transactions on Image Processing* 23.12, pp. 5348–5359.

Mantiuk, Rafal K., Karol Myszkowski, and Hans-Peter Seidel (2015). "High Dynamic Range Imaging." In: *Wiley Encyclopedia of Electrical and Electronics Engineering*. John Wiley & Sons, Inc.

Mao, Xiaojiao, Chunhua Shen, and Yu-Bin Yang (2016). "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections." In: *Advances in Neural Information Processing Systems*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, pp. 2802–2810.

Marĉelja, S (1980). "Mathematical description of the responses of simple cortical cells." In: *Journal of the Optical Society of America* 70.11, pp. 1297–1300.

Marco, Daniel and David L. Neuhoff (2005). "The validity of the additive noise model for uniform scalar quantizers." In: *IEEE Transactions on Information Theory* 51.5, pp. 1739–1755.

Mardia, Kanti V. and Peter E. Jupp (2009). *Directional statistics*. John Wiley & Sons.

Martin, David, Charless Fowlkes, Doron Tal, and Jitendra Malik (July 2001a). "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." In: *Proceedings of the Eighth IEEE International Conference on Computer Vision*. Vol. 2. Vancouver, British Columbia, Canada: IEEE, pp. 416–423.

Martin, David, Charless Fowlkes, Doron Tal, and Jitendra Malik (July 2001b). "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics." In: *Proceedings of the Eighth IEEE International*

Conference on Computer Vision. Vol. 2. Vancouver, British Columbia, Canada, pp. 416–423.

Meer, Peter, Jean-Michel Jolion, and Azriel Rosenfeld (1990). "A fast parallel algorithm for blind estimation of noise variance." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.2, pp. 216–223.

Mei, Jonathan, Ahmed Kirmani, Andrea Colaço, and Vivek K. Goyal (Sept. 2013). "Phase unwrapping and denoising for time-of-flight imaging using generalized approximate message passing." In: *Proceedings of the IEEE International Conference on Image Processing*. Melbourne, Australia, pp. 364–368.

Metzler, Christopher A, Ali Mousavi, Reinhard Heckel, and Richard G Baraniuk (2018). "Unsupervised Learning with Stein's Unbiased Risk Estimator." In: *arXiv:1805.10531 [stat.ML]*.

Miskin, James and David J. C. MacKay (2000). "Ensemble Learning for Blind Image Separation and Deconvolution." In: *Advances in Independent Component Analysis*. Ed. by Mark Girolami. Perspectives in Neural Computing. Springer London. Chap. 7, pp. 123–141. ISBN: 978-1-4471-0443-8.

Mittal, Anish, Anush Krishna Moorthy, and Alan Conrad Bovik (2012). "No-reference image quality assessment in the spatial domain." In: *IEEE Transactions on Image Processing* 21.12, pp. 4695–4708.

Mnih, Andriy and Danilo J. Rezende (June 2016a). "Variational Inference for Monte Carlo Objectives." In: *Proceedings of the 33rd International Conference on Machine Learning*. New York, NY, pp. 2188–2196.

Mnih, Andriy and Danilo Jimenez Rezende (June 2016b). "Variational inference for Monte Carlo objectives." In: *Proceedings of the 33rd International Conference on Machine Learning*. New York, NY: PMLR, pp. 2188–2196.

Moldovan, Teodor Mihai, Stefan Roth, and Michael J. Black (Oct. 2006). "Denoising Archival Films using a Learned Bayesian Model." In: *Proceedings of the IEEE International Conference on Image Processing*. Atlanta, Georgia, pp. 2641–2644.

Moorthy, Anush Krishna and Alan Conrad Bovik (2011). "Blind image quality assessment: From natural scene statistics to perceptual quality." In: *IEEE Transactions on Image Processing* 20.12, pp. 3350–3364.

Mosseri, Inbar, Maria Zontak, and Michal Irani (Apr. 2013). "Combining the power of internal and external denoising." In: *IEEE International Conference on Computational Photography (ICCP)*. Cambridge, Massachusetts, USA.

Nam, Seonghyeon, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim (June 2016). "A Holistic Approach to Cross-Channel Image Noise Modeling and Its Application to Image Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 1683–1691.

Nayar, Shree K. and Tomoo Mitsunaga (June 2000). "High dynamic range imaging: Spatially varying pixel exposures." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Hilton Head Island, South Carolina, pp. 472–479.

Neal, Radford M. (2001). "Annealed importance sampling." In: *Statistics and Computing* 11.2, pp. 125–139.

Nikolova, Mila and Raymond H. Chan (June 2007). "The Equivalence of Half-Quadratic Minimization and the Gradient Linearization Iteration." In: *IEEE Transactions on Image Processing* 16.6, pp. 1623–1627.

Nyström, Evert Johannes (Sept. 1928). "Über die Praktische Lösung von linearen integralgleichungen mit anwendungen auf Randwertaufgaben der potentialtheorie." In: *Commentationes Physico-Mathematicae* 4.15, pp. 1–52.

Ochs, Peter, Jitendra Malik, and Thomas Brox (June 2014). "Segmentation of Moving Objects by Long Term Video Analysis." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.6, pp. 1187–1200.

Olshausen, B[runo] A. and D[avid] J. Field (May 1996). "Natural Image Statistics and Efficient Coding." In: *Network: Computation in Neural Systems* 7.2, pp. 333–339.

Olshausen, Bruno A. and David J. Field (Dec. 1997). "Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?" In: *Vision Research* 37.23, pp. 3311–3325.

Olshausen, Bruno A. and K. Jarrod Millman (2000). "Learning Sparse Codes with a Mixture-of-Gaussians Prior." In: *Advances in Neural Information Processing Systems*. Ed. by S. A. Solla, T. K. Leen, and K.-R. Müller. Vol. 12, pp. 841–847.

Ono, Shunsuke (2017). "Primal-dual plug-and-play image restoration." In: *IEEE Signal Processing Letters* 24.8, pp. 1108–1112.

Oord, Aaron van den, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. (2016). "Conditional image generation with PixelCNN decoders." In: *Advances in Neural Information Processing Systems*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, pp. 4790–4798.

Oppenheim, Alan V. and Ronald Schafer (1975). *Digital Signal Processing*. Prentice Hall.

Parikh, Neal and Stephen Boyd (2014). "Proximal algorithms." In: *Foundations and Trends in Optimization* 1.3, pp. 127–239.

Park, Sung Hee, Hyung Suk Kim, Steven Lansel, Manu Parmar, and Brian A Wandell (2009). "A case for denoising before demosaicking color filter array data." In: *Asilomar Conf. on Signals, Systems and Computers*, pp. 860–864.

Perona, Pietro and Jitendra Malik (July 1990). "Scale-Space and Edge Detection using Anisotropic Diffusion." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7, pp. 629–639.

Plötz, Tobias and Stefan Roth (July 2017). "Benchmarking Denoising Algorithms with Real Photographs." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 1586–1595.

Plötz, Tobias and Stefan Roth (2018). "Neural Nearest Neighbors Networks." In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31, pp. 1087–1098.

Plötz, Tobias, Anne S. Wannenwetsch, and Stefan Roth (June 2018). "Stochastic Variational Inference with Gradient Linearization." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 1566–1575.

Portilla, Javier (Oct. 2004). "Full Blind Denoising through Noise Covariance Estimation using Gaussian Scale Mixtures in the Wavelet Domain." In: *Proceedings of the IEEE International Conference on Image Processing*. Vol. 3. Singapore, pp. 1217–1220.

Portilla, Javier and Eero P. Simoncelli (Sept. 2000). "Image Denoising via Adjustment of Wavelet Coefficients Magnitude Correlation."

In: *Proceedings of the 7th International Conference on Image Processing*. Vol. 3. Vancouver, Canada, pp. 277–280.

Portilla, Javier, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli (Nov. 2003). "Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain." In: *IEEE Transactions on Image Processing* 12.11, pp. 1338–1351.

Pyatykh, Stanislav, Jürgen Hesser, and Lei Zheng (2013). "Image noise level estimation by principal component analysis." In: *IEEE Transactions on Image Processing* 22.2, pp. 687–699.

Ramani, Sathish, Thierry Blu, and Michael Unser (2008). "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms." In: *IEEE Transactions on Image Processing* 17.9, pp. 1540–1554.

Ranganath, Rajesh, Sean Gerrish, and David M. Blei (Apr. 2014). "Black Box Variational Inference." In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*. Reykjavik, Iceland: PMLR, pp. 814–822.

Rank, Klaus, Markus Lendl, and Rolf Unbehauen (1999). "Estimation of image noise variance." In: *IEE Proceedings - Vision, Image and Signal Processing* 146.2, pp. 80–84.

Rastegari, Mohammad, Vicente Ordonez, Joseph Redmon, and Ali Farhadi (2016). "XNOR-Net: Imagenet classification using binary convolutional neural networks." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9908. Lecture Notes in Computer Science. Springer, pp. 525–542.

Ren, Weiqiang, Yinan Yu, Junge Zhang, and Kaiqi Huang (Aug. 2014). "Learning convolutional nonlinear features for k nearest neighbor image classification." In: *Proceedings of 22nd IEEE International Conference on Pattern Recognition*. Stockholm, Sweden, pp. 4358–4363.

Revaud, Jerome, Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid (June 2015). "EpicFlow: Edge-Preserving Interpolation of Correspondences for Optical Flow." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Boston, Massachusetts: IEEE.

Rezende, Danilo J., Shakir Mohamed, and Daan Wierstra (June 2014). "Stochastic Backpropagation and Approximate Inference in Deep

Generative Models." In: *Proceedings of the 31st International Conference on Machine Learning*. Beijing, China: PMLR, pp. 1278–1286.

Robbins, Herbert and Sutton Monro (1951). "A stochastic approximation method." In: *The Annals of Mathematical Statistics* 22.3, pp. 400–407.

Romano, Yaniv, Michael Elad, and Peyman Milanfar (2017). "The little engine that could: Regularization by denoising (RED)." In: *SIAM Journal on Imaging Sciences* 10.4, pp. 1804–1844.

Roth, Stefan and Michael J. Black (June 2005). "Fields of Experts: A Framework for Learning Image Priors." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Vol. 2. San Diego, California, pp. 860–867.

Roth, Stefan and Michael J. Black (Apr. 2009). "Fields of Experts." In: *International Journal of Computer Vision* 82.2, pp. 205–229. extended version of (Roth and Black, 2005).

Roth, Stefan and Michael J. Black (2011). "Fields of Experts." In: *Advances in Markov Random Fields for Vision and Image Processing*. Ed. by Andrew Blake, Pushmeet Kohli, and Carsten Rother. MIT Press. Chap. 19.

Ruderman, Daniel L. (Nov. 1994). "The Statistics of Natural Images." In: *Network: Computation in Neural Systems* 5.4, pp. 517–548.

Ruderman, Daniel L. (Dec. 1997). "Origins of Scaling in Natural Images." In: *Vision Research* 37.23, pp. 3385–3398.

Ruiz, Francisco R., Michalis K. Titsias, and David M. Blei (2016). "The generalized reparameterization gradient." In: *Advances in Neural Information Processing Systems*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett. Vol. 29, pp. 460–468.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. (2015). "Imagenet large scale visual recognition challenge." In: *International Journal of Computer Vision* 115.3, pp. 211–252.

Saad, Michele A., Alan C. Bovik, and Christophe Charrier (2012). "Blind image quality assessment: A natural scene statistics approach in the DCT domain." In: *IEEE Transactions on Image Processing* 21.8, pp. 3339–3352.

Sajjadi, Mehdi SM, Bernhard Schölkopf, and Michael Hirsch (Oct. 2017). "EnhanceNet: Single image super-resolution through automated texture synthesis." In: pp. 4501–4510.

Schelten, Kevin and Stefan Roth (June 2011). "Connecting Non-Quadratic Variational Models and MRFs." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado Springs, Colorado, pp. 2641–2648.

Schelten, Kevin and Stefan Roth (2012). "Mean Field for Continuous High-Order MRFs." In: *Pattern Recognition, Proceedings of the 34th DAGM-Symposium*. Ed. by A. Pinz, T. Pock, H. Bischof, and F. Leberl. Vol. 7476. Lecture Notes in Computer Science. Springer, pp. 52–61.

Schmidt, Uwe and Stefan Roth (June 2014). "Shrinkage Fields for Effective Image Restoration." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio, pp. 2774–2781.

Schmidt, Uwe, Qi Gao, and Stefan Roth (June 2010). "A Generative Perspective on MRFs in Low-Level Vision." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Francisco, California, pp. 1751–1758.

Schmidt, Uwe, Kevin Schelten, and Stefan Roth (June 2011). "Bayesian Deblurring with Integrated Noise Estimation." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Colorado Springs, Colorado, pp. 2625–2632.

Schölkopf, Bernhard, Alexander J. Smola, Francis Bach, et al. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*.

Sheikh, Hamid R. and Alan C. Bovik (2006). "Image information and visual quality." In: *IEEE Transactions on Image Processing* 15.2, pp. 430–444.

Sheikh, Hamid R., Alan C. Bovik, and Gustavo De Veciana (2005). "An information fidelity criterion for image quality assessment using natural scene statistics." In: *IEEE Transactions on Image Processing* 14.12, pp. 2117–2128.

Simoncelli, Eero P. (Nov. 1997). "Statistical Models for Images: Compression, Restoration and Synthesis." In: *Proceedings of the 31st Asilomar Conference on Signals, Systems, and Computers*. Vol. 1. Pacafic Grove, California, pp. 673–678.

Simoncelli, Eero P. (1999). "Bayesian Denoising of Visual Images in the Wavelet Domain." In: *Bayesian Inference in Wavelet Based Models*. Ed. by P. Müller and B. Vidakovic. Vol. 141. Lecture Notes in Statistics. Springer. Chap. 18, pp. 292–308.

Simoncelli, Eero P and Edward H Adelson (Sept. 1996). "Noise removal via Bayesian wavelet coring." In: *Proceedings of 3rd IEEE International Conference on Image Processing (ICIP)*. Lausanne, Switzerland, pp. 379–382.

Smith, Stephen M. and J. Michael Brady (1997). "SUSAN — a new approach to low level image processing." In: *International Journal of Computer Vision* 23.1, pp. 45–78.

Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams (2012). "Practical Bayesian Optimization of Machine Learning Algorithms." In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger. Vol. 25, pp. 2951–2959.

Soltanayev, Shakarim and Se Young Chun (2018). "Training Deep Learning based Denoisers without Ground Truth Data." In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, pp. 3257–3267.

Starck, Jean-Luc, Fionn Murtagh, and Albert Bijaoui (1998). *Image Processing and Data Analysis: The Multiscale Approach*. Cambridge University Press.

Stein, Charles M. (1981). "Estimation of the mean of a multivariate normal distribution." In: *The annals of statistics*, pp. 1135–1151.

Su, Shuochen, Felix Heide, Gordon Wetzstein, and Wolfgang Heidrich (June 2018). "Deep End-to-End Time-of-Flight Imaging." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 6383–6392.

Tai, Ying, Jian Yang, Xiaoming Liu, and Chunyan Xu (Oct. 2017). "MemNet: A Persistent Memory Network for Image Restoration." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 4539–4547.

Talebi, Hossein and Peyman Milanfar (2014). "Global image denoising." In: *IEEE Transactions on Image Processing* 23.2, pp. 755–768.

Tappen, Marshall, Ce Liu, Edward H. Adelson, and William T. Freeman (June 2007). "Learning Gaussian Conditional Random Fields

for Low-Level Vision." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota.

Teh, Yee Whye, Kenichi Kurihara, and Max Welling (2008). "Collapsed variational inference for HDP." In: *Advances in Neural Information Processing Systems*. Ed. by J. C. Platt, D. Koller, Y. Singer, and S. Roweis. Vol. 20, pp. 1481–1488.

Tieleman, Tijmen and Geoffrey E. Hinton (2012). *Lecture 6.5 – RMSprop: Divide the gradient by a running average of its recent magnitude*. Tech. rep. COURSERA: Neural networks for machine learning.

Timofte, Radu, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, and Lei Zhang (July 2017). "NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results." In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New Trends in Image Restoration and Enhancement Workshop (NTIRE)*. Honolulu, Hawaii, pp. 114–125.

Tobin, James (Jan. 1958). "Estimation of Relationships for Limited Dependent Variables." In: *Econometrica* 26.1, pp. 24–36.

Tomasi, C[arlo] and R. Manduchi (Jan. 1998). "Bilateral Filtering for Gray and Color Images." In: *Proceedings of the Sixth IEEE International Conference on Computer Vision*. Bombay, India, pp. 839–846.

Tran, Dustin, Matthew D. Hoffman, Rif A. Saurous, Eugene Brevdo, Kevin Murphy, and David M. Blei (2017). "Deep probabilistic programming." In: *Proceedings of the International Conference on Learning Representations*, Toulon, France.

Turcotte, Donald L (1995). "Scaling in geology: Landforms and earthquakes." In: *Proceedings of the National Academy of Sciences* 92.15, pp. 6697–6704.

Valadao, Gonçalo and José Bioucas-Dias (Aug. 2008). "Phase imaging: Unwrapping and denoising with diversity and multi-resolution." In: *Proceedings of 2008 International Workshop on Local and Non-Local Approximation in Image Processing (LNLA'08)*. Lausanne, Switzerland.

Van De Ville, Dimitri and Michel Kocher (2009). "SURE-Based Non-Local Means." In: *IEEE Signal Processing Letters* 16.11, pp. 973–976.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin (2017). "Attention is all you need." In: *Advances in Neural Information Process-*

ing Systems. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, pp. 6000–6010.

Venkatakrishnan, Singanallur V., Charles A. Bouman, and Brendt Wohlberg (2013). "Plug-and-play priors for model based reconstruction." In: *Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, pp. 945–948.

Vinyals, Oriol, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra (2016). "Matching networks for one shot learning." In: *Advances in Neural Information Processing Systems*. Ed. by D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, pp. 3630–3638.

Vogel, Curtis R. and Mary E. Oman (1996). "Iterative Methods for Total Variation Denoising." In: *SIAM Journal on Scientific Computing* 17.1, pp. 227–238.

Vogel, Curtis R. and Mary E. Oman (1998). "Fast, Robust Total Variation-Based Reconstruction of Noisy, Blurred Images." In: *IEEE Transactions on Image Processing* 7.6, pp. 813–824.

Wainwright, Martin J. and Michael I. Jordan (Jan. 2008). "Graphical Models, Exponential Families, and Variational Inference." In: *Foundations and Trends in Machine Learning* 1.1–2, pp. 1–305.

Wang, Xiaolong, Ross Girshick, Abhinav Gupta, and Kaiming He (June 2018). "Non-local Neural Networks." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 7794–7803.

Wang, Zhou, Eero P. Simoncelli, and Alan C. Bovik (Nov. 2003). "Multiscale Structural Similarity for Image Quality Assessment." In: *Proceedings of the 37th IEEE Signal Processing Society Asilomar Conference on Signals, Systems and Computers*. Pacific Grove, California, pp. 1398–1402.

Wang, Zhou, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli (Apr. 2004). "Image Quality Assessment: From Error Visibility to Structural Similarity." In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612.

Wannenwetsch, Anne S., Magret Keuper, and Stefan Roth (Oct. 2017). "ProbFlow: Joint Optical Flow and Uncertainty Estimation." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 1182 –1191.

Wedel, Andreas, Annemarie Meißner, Clemens Rabe, Uwe Franke, and Daniel Cremers (2009). "Detection and Segmentation of Independently Moving Objects from Dense Scene Flow." In: *Proceedings of the 7th International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Bonn, Germany: Springer, pp. 14–27.

Weinberger, Kilian Q. and Lawrence K. Saul (Feb. 2009). "Distance metric learning for large margin nearest neighbor classification." In: *Journal of Machine Learning Research* 10, pp. 207–244.

Weiss, Yair and William T. Freeman (June 2007). "What Makes a Good Model of Natural Images?" In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Minneapolis, Minnesota.

Winn, John and Christopher M. Bishop (Apr. 2005). "Variational Message Passing." In: *Journal of Machine Learning Research* 6, pp. 661–694.

Wu, Yuxin and Kaiming He (2018). "Group normalization." In: Lecture Notes in Computer Science 11217. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, pp. 3–19.

Xiao, Lei, Felix Heide, Wolfgang Heidrich, Bernhard Schölkopf, and Michael Hirsch (2018). "Discriminative Transfer Learning for General Image Restoration." In: *IEEE Transactions on Image Processing* 27.8, pp. 4091–4104.

Xu, Huijuan and Kate Saenko (2016). "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering." In: *Proceedings of the 14th European Conference on Computer Vision*. Ed. by B. Leibe, J. Matas, N. Sebe, and M. Welling. Vol. 9906. Lecture Notes in Computer Science. Springer, pp. 451–466.

Xu, Jun, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng (Dec. 2015). "Patch group based nonlocal self-similarity prior learning for image denoising." In: *Proceedings of the Fifteenth IEEE International Conference on Computer Vision*. Santiago, Chile, pp. 244–252.

Xu, Jun, Lei Zhang, David Zhang, and Xiangchu Feng (Oct. 2017a). "Multi-channel weighted nuclear norm minimization for real color image denoising." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 1096–1104.

Xu, Jun, Lei Zhang, and David Zhang (2018a). "A Trilateral Weighted Sparse Coding Scheme for Real-World Image Denoising." In: *Proceedings of the 15th European Conference on Computer Vision*. Ed. by V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss. Vol. 11212. Lecture Notes in Computer Science. Springer, pp. 21–38.

Xu, Jun, Lei Zhang, and David Zhang (2018b). "External Prior Guided Internal Prior Learning for Real-World Noisy Image Denoising." In: *IEEE Transactions on Image Processing* 27.6, pp. 2996–3010.

Xu, Xiangyu, Deqing Sun, Jinshan Pan, Yujin Zhang, Hanspeter Pfister, and Ming-Hsuan Yang (Oct. 2017b). "Learning to super-resolve blurry face and text images." In: *Proceedings of the Sixteenth IEEE International Conference on Computer Vision*. Venice, Italy, pp. 251–260.

Yang, Dong and Jian Sun (2018). "BM3D-Net: A Convolutional Neural Network for Transform-Domain Collaborative Filtering." In: *IEEE Transactions on Signal Processing* 25.1, pp. 55–59.

Yang, Jianchao, John Wright, Thomas S. Huang, and Yi Ma (2010). "Image super-resolution via sparse representation." In: *IEEE Transactions on Image Processing* 19.11, pp. 2861–2873.

Yi, Kwang Moo, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua (June 2018). "Learning to Find Good Correspondences." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 2666–2674.

Young, David Matheson (1971). *Iterative solution of large linear systems*. New York: Academic Press.

Yu, Fisher and Vladlen Koltun (2015). "Multi-scale context aggregation by dilated convolutions." In: *ICLR*.

Yu, Ke, Chao Dong, Liang Lin, and Chen Change Loy (June 2018). "Crafting a toolchain for image restoration by deep reinforcement learning." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Salt Lake City, Utah, pp. 2443–2452.

Zhang, Jiachao, Keigo Hirakawa, and Xiaodan Jin (Apr. 2015). "Quantile analysis of image sensor noise distribution." In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. South Brisbane, Australia, pp. 1598–1602.

Zhang, Kai, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang (2017a). "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising." In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155.

Zhang, Kai, Wangmeng Zuo, Shuhang Gu, and Lei Zhang (July 2017b). "Learning deep CNN denoiser prior for image restoration." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Honolulu, Hawaii, pp. 2808–2817.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2018). "FFDNet: Toward a fast and flexible solution for CNN based image denoising." In: *IEEE Transactions on Image Processing* 27.9, pp. 4608–4622.

Zhang, Shuangteng and Ezzatollah Salari (Mar. 2005). "Image denoising using a neural network based non-linear filter in wavelet domain." In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. Philadelphia, Pennsylvania, pp. 989–992.

Zhang, Xiaoshuai, Yiping Lu, Jiaying Liu, and Bin Dong (May 2019). "Dynamically Unfolding Recurrent Restorer: A Moving Endpoint Control Method for Image Restoration." In: *Proceedings of the Seventh International Conference on Learning Representations*. New Orleans, Louisiana.

Zhao, Hang, Boxin Shi, Christy Fernandez-Cull, Sai-Kit Yeung, and Ramesh Raskar (Apr. 2015). "Unbounded High Dynamic Range Photography using a Modulo Camera." In: *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*. Houston, Texas, USA.

Zhao, Hong, Wenyi Chen, and Yushan Tan (1994). "Phase-unwrapping algorithm for the measurement of three-dimensional object shapes." In: *OSA Applied Optics* 33.20, pp. 4497–4500.

Zhu, Fengyuan, Guangyong Chen, and Pheng-Ann Heng (June 2016). "From Noise Modeling to Blind Image Denoising." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Las Vegas, Nevada, pp. 420–429.

Zhu, Song Chun and David Mumford (Nov. 1997). "Prior Learning and Gibbs Reaction-Diffusion." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.11, pp. 1236–1250.

Zontak, Maria and Michal Irani (June 2011). "Internal statistics of a single natural image." In: *Proceedings of the IEEE Computer Soci-*

ety Conference on Computer Vision and Pattern Recognition. Colorado Springs, Colorado, pp. 977–984.

Zontak, Maria, Inbar Mosseri, and Michal Irani (June 2013). "Separating signal from noise using patch recurrence across scales." In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Portland, Oregon, pp. 1195–1202.

Zoran, Daniel and Yair Weiss (Oct. 2009). "Scale Invariance and Noise in Natural Images." In: *Proceedings of the Twelfth IEEE International Conference on Computer Vision*. Kyoto, Japan, pp. 2209–2216.

Zoran, Daniel and Yair Weiss (Nov. 2011). "From Learning Models of Natural Image Patches to Whole Image Restoration." In: *Proceedings of the Thirteenth IEEE International Conference on Computer Vision*. Barcelona, Spain, pp. 479–486.

Zuo, Chao, Lei Huang, Minliang Zhang, Qian Chen, and Anand Asundi (2016). "Temporal phase unwrapping algorithms for fringe projection profilometry: A comparative review." In: *Elsevier Optics and Lasers in Engineering* 85, pp. 84–103.

CURRICULUM VITÆ

TOBIAS PLÖTZ

Education	2014 – 2019	*Technische Universität Darmstadt, Germany*
		Ph.D. student in Computer Science
	2011 – 2014	*Technische Universität Darmstadt, Germany*
		M.Sc. in Computer Science
	2008 – 2011	*Technische Universität Darmstadt, Germany*
		B.Sc. in Computer Science
	1999 – 2007	*Ernst-Ludwig-Schule, Bad Nauheim, Germany*

Positions	Since 2019	*Merck KGaA Darmstadt, Germany*
		Global data science
		Data Scientist, Life Science
	2014 – 2019	*Technische Universität Darmstadt, Germany*
		Visual Inference group of Prof. Stefan Roth
		Research and teaching assistant
	2010 – 2014	*Technische Universität Darmstadt, Germany*
		Student research assistant

PUBLICATIONS

Tobias Plötz and Stefan Roth
Neural nearest neighbors networks. *Advances in Neural Information Processing Systems (NeurIPS)*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Editors, volume 31, pages 1087–1098, December 2018.

Tobias Plötz, Anne S. Wannenwetsch, and Stefan Roth
Stochastic variational inference with gradient linearization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, pages 1566–1575, June 2018.

Florian Lang, Tobias Plötz, and Stefan Roth
Robust multi-image HDR reconstruction for the modulo camera. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, V. Roth and T. Vetter, Editors, series LNCS, volume 10496, pages 78–89, Springer, 2017.

Tobias Plötz and Stefan Roth
Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, Hawaii, pages 1586-1595, July 2017.

Tobias Plötz and Stefan Roth
Automatic registration of images to untextured geometry using average shading gradients. *International Journal of Computer Vision (IJCV)*, volume 125, number 1–3, pages 65–81, 2017.

Tobias Plötz and Stefan Roth
Registering images to untextured geometry using average shading gradients. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pages 2030–2038, December 2015.