

# Real-Time Noise Source Estimation of a Camera System from an Image and Metadata

Maik Wischow,\* Patrick Irmisch, Anko Boerner, and Guillermo Gallego

Autonomous machines must self-maintain proper functionality to ensure the safety of humans and themselves. This pertains particularly to its cameras as predominant sensors to perceive the environment and support actions. A fundamental camera problem addressed in this study is noise. Solutions often focus on denoising images a posteriori, that is, fighting symptoms rather than root causes. However, tackling root causes requires identifying the noise sources, considering the limitations of mobile platforms. In this work, a real-time, memory-efficient, and reliable noise source estimator that combines data-based and physically based models is investigated. To this end, a deep neural network that examines an image with camera metadata for major camera noise sources is built and trained. In addition, it quantifies unexpected factors that impact image noise or metadata. This study investigates seven different estimators on six datasets that include synthetic noise, real-world noise from two camera systems, and real-field campaigns. For these, only the model with most metadata is capable to accurately and robustly quantify all individual noise contributions. This method outperforms total image noise estimators and can be plug-and-play deployed. It also serves as a basis to include more advanced noise sources, or as part of an automatic countermeasure feedback loop to approach fully reliable machines.

## 1. Introduction

Machines in various fields (e.g., vehicles, robots) are increasingly moving away from manual control toward autonomy, which implies that they should ensure proper operation (e.g.,<sup>[1–3]</sup>). This applies to each component of a machine, and in particular to its perception system, as all subsequent actions depend on it. Cameras are the predominant sensors for perceiving the environment and are therefore the subject of our study. As any physical sensor, a camera is afflicted with noise, whose influence on subsequent computer vision tasks has justified extensive research. To guarantee a machine's dependability and durability, which in turn guarantees the safety of both humans and machines, counteracting noise is mandatory. However, to counteract noise in an active system, one needs first to identify and quantify its root causes.

Previous studies approach this task by using estimated noise levels to denoise images.<sup>[4–8]</sup> This process has matured for various noise models and use cases, but

they often yield undesired visual artifacts. That is, they only fight symptoms and do not target noise source identification, although noise sources and countermeasures are well researched,<sup>[9]</sup> Section 7. This can be attributed to three reasons: 1) the camera system control is often inaccessible, which makes denoising more applicable if only image datasets are available. 2) The need for more autonomy of machines with consumer-grade cameras emerged only recently and noise could only be approached manually so far. 3) Reliable and real-time noise source estimation is challenging; it relies on accurate image noise estimation and extensive noise models, which gained interest and matured only in recent years (see Section 2). Moreover, noise source identification from an image alone is ambiguous, since most noise sources follow similar statistics, auxiliary data is needed for disambiguation. Last but not least, at present, only deep neural networks (DNNs) are able to perform the implied complex operations (extensive noise modeling and heterogeneous data fusion) in real time.


This article proposes a real-time, memory-efficient, and reliable noise source estimator (**Figure 1**). During operation, it analyzes single images together with metadata from the camera system and quantifies the respective contributions to major noise sources of the system. Moreover, we include a verification mechanism that quantifies noise mismatches between the metadata

M. Wischow, P. Irmisch, A. Boerner  
Department of Real-Time Data Processing, Institute of Optical Sensor Systems  
German Aerospace Center  
Berlin 12489, Germany  
E-mail: Maik.Wischow@gmx.de

M. Wischow, G. Gallego  
Department of Electrical Engineering and Computer Science  
Technische Universität Berlin  
10623 Berlin, Germany

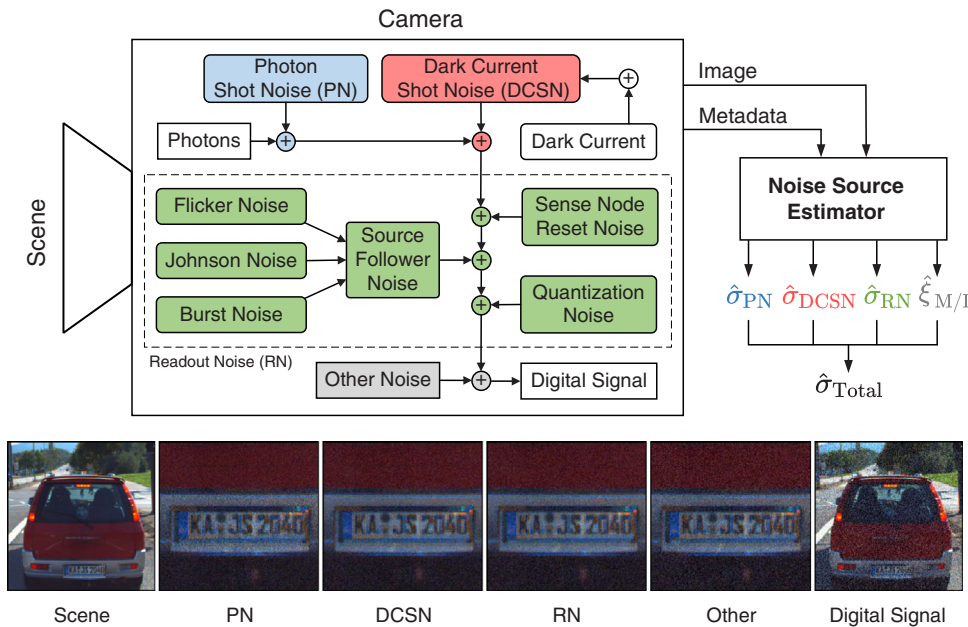
G. Gallego  
Einstein Center Digital Future  
10117 Berlin, Germany

G. Gallego  
Science of Intelligence Excellence Cluster  
10587 Berlin, Germany

 The ORCID identification number(s) for the author(s) of this article can be found under <https://doi.org/10.1002/aisy.202300479>.

© 2024 The Authors. Advanced Intelligent Systems published by Wiley-VCH GmbH. This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

DOI: 10.1002/aisy.202300479



**Figure 1.** Proposed camera noise source estimation. Different noise sources affect the image formation process of a scene. Our noise source estimator quantifies major noise source contributions  $\hat{\sigma}_{i \in \{PN, DCSN, RN\}}$ , unexpected noise  $\hat{\zeta}_{M/I}$ , and the total image noise  $\hat{\sigma}_{Total}$  using an image and camera metadata.

and the image noise, which serves for self-control and detection of unexpected events (e.g., camera damages). Without loss of generality, our study analyzes: time-varying noise (since any time-invariant noise is usually mitigated by camera calibration), and spatially varying noise (since image patches are used).

We make the following technical contributions: 1) we propose a real-time, memory-efficient, and reliable DNN-based noise source estimator (Section 3) that is able to quantify contributions of different camera noise sources and to detect unexpected factors that impact image noise or metadata. 2) We demonstrate seven different estimators in comprehensive experiments on six datasets and two real camera systems (Section 4). Our experiments investigate synthetic noise, real-world noise extracted from camera systems, and qualitative field campaigns, and also create unexpected noise events in images or metadata. 3) We provide the source code of our experiments, the data used for training and benchmarking, and the ready-to-use estimators (<https://github.com/MaikWischow/Noise-Source-Estimation>).

## 2. Related Work

We first survey general image noise level estimators and then discuss noise models from related studies that account for multiple noise sources and utilize camera metadata.

### 2.1. Noise Level Estimation

Motivated by applications in the field, we focus our study on estimators that assume unknown noise levels (i.e., blind estimation) using single images. These may be further divided into traditional and learning-based approaches. Traditional approaches comprise one or more of the following paradigms: 1) block-based<sup>[10,11]</sup>

(estimate noise using low-textured regions), 2) filtering-based<sup>[10,12]</sup> (subtract a low-pass filtered image and estimate noise from high frequency components), and 3) transform-based<sup>[4,13,14]</sup> (represent the image in a different space, e.g., using wavelets, and estimate noise therein). All have their own pros and cons, with over-/underestimation in low/high noise and textured areas.

Learning-based methods either determine the noise level explicitly<sup>[6,8,15]</sup> (e.g., using residual learning and scale pyramids) or implicitly<sup>[5,16]</sup> (e.g., with generative adversarial networks) as part of an end-to-end denoising pipeline. In terms of real-image denoising performance, traditional methods are still considered the state of the art, closely followed by learning-based methods.<sup>[17]</sup>

### 2.2. Noise Models

Driven by space camera systems, extensive noise models on a subatomic level have been developed in recent decades.<sup>[9,18,19]</sup> However, applications on earth tend to employ simpler models, as follows.

The majority of research presumes an additive white Gaussian noise source.<sup>[4,5,8,10–14]</sup> Given the influence of light on camera noise,<sup>[20]</sup> signal-dependent noise models have been developed considering 1) photon shot noise and 2) noise due to camera electronics (e.g., the Poissonian–Gaussian noise model).<sup>[6,15,21]</sup> A special case is the noise level function (NLF) that characterizes the dependence of noise levels on image intensity.<sup>[22–24]</sup> To account for nonlinear camera processes that affect noise statistics,<sup>[25]</sup> some works employ the camera response function for NLF estimation; they describe a camera's physical processing as a black box in a single function.<sup>[26,27]</sup>

A few noise studies break down the noise caused by camera electronics and therefore consider more than two noise

sources,<sup>[7,28–30]</sup> but generally “[...] noise sources caused by digital camera electronics are still largely overlooked, despite their significant effect on raw measurement”.<sup>[28]</sup> The works in refs. [28,30] propose “simpler” extensive noise models that account, e.g., for the camera system gain, read noise, or quantization noise, which are partially analyzed in more detail. More sophisticated noise models from refs. [7,29] also address camera specifics like the shutter mechanism, individual color channel biases, or differentiate between analog/digital gain. There have also been attempts to approximate noise models by DNNs<sup>[31–34]</sup> for synthesis, but<sup>[29]</sup> shows that “The DNN-based [noise generators] still cannot outperform physics-based statistical methods”.

All the aforementioned models calibrate their parameters (temperature, exposure time, International Organization for Standardization gain, ...) offline and only implicitly account for changing camera parameters during training data generation, but they do not consider camera parameters at inference time. We investigate this gap and show in our experiments on DNN noise level estimation that the system is only able to identify the contribution of different noise sources when these parameters are available at runtime. Furthermore, to the best of our knowledge, our approach is the first estimator (traditional or learning based) to explicitly quantify not only two (photon shot noise [PN], other) but four (PN, dark current shot noise [DCSN], readout noise [RN], other) individual noise source contributions.

### 3. Noise Source Estimation

Given a possibly corrupted image patch  $I'$  and metadata from the camera system, the goal of our image noise source estimator is to determine the image’s total noise level:

$$\sigma_{\text{Total}} = \sqrt{\sigma_{\text{PN}}^2 + \sigma_{\text{DCSN}}^2 + \sigma_{\text{RN}}^2 + \xi_{\text{M/I}}} \quad (1)$$

and its individual components: the PN level  $\sigma_{\text{PN}}$ , the DCSN level  $\sigma_{\text{DCSN}}$ , the RN level  $\sigma_{\text{RN}}$ , and a component  $\xi_{\text{M/I}}$  that quantifies unexpected (i.e., residual) noise (details about noise types are in Supporting Information). We assume grayscale patches, of size  $128 \times 128$  px. Next, we describe the base architecture (Section 3.1), subsequently detail our extensions (Section 3.2), and lastly focus on training the noise source estimator (Section 3.3).

#### 3.1. Base Architecture

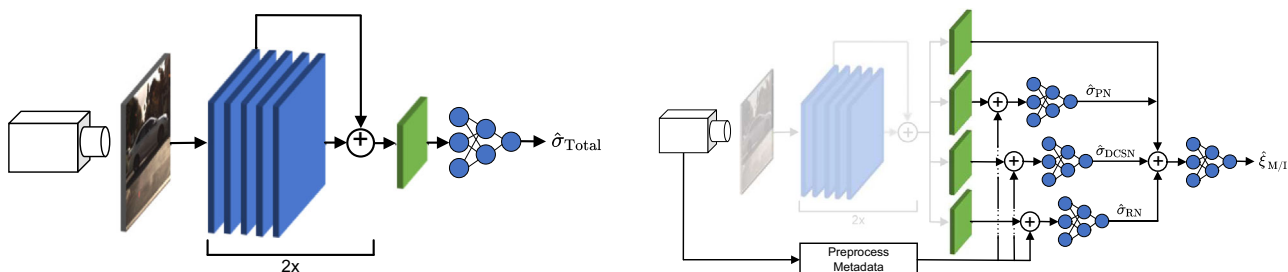
Our method is inspired by the deep residual noise-level estimator (DRNE) from ref. [6], which has been shown to be superior compared to traditional state-of-the-art approaches in terms of runtime and accuracy.<sup>[35]</sup> It takes an red-green-blue (RGB) image patch as input and predicts a pixel-wise noise level. It consists of 16 convolution layers (with 15 of them separated into three residual blocks). Pooling layers, interpolation operations, and convolution strides larger than  $3 \times 3$  are omitted to keep the focus on low-level noise features.

We customize the aforementioned architecture so that the neural network takes grayscale images as input and estimates only one noise level per image patch (left part of **Figure 2**). Specifically, we replace the first  $3 \times 3 \times 3$  convolution kernel by a  $3 \times 3$  one, replace the last residual block by a fully connected block (FCB) with three layers having 32, 16, and 8 neurons, respectively, and apply global max pooling before the FCB to fit the dimensions. As a consequence, we are able to reduce the total number of network parameters by 35%, from 519 to 336 k while achieving similar estimation accuracy as ref. [6]. Lastly, we retrain the network as described in Section 3.3. In the upcoming sections, we refer to this customized model as  $\text{DRNE}_{\text{cust.}}$ .

#### 3.2. Noise Source Estimation

The previous method estimates the noise level of the patch, but does not identify its origin (i.e., type and amount of noise), which is critical information for a camera’s maintenance operation. To identify the noise origin, additional information is needed alongside the noised image. Our approach is to train the baseline network on a physical noise model<sup>[19]</sup> that relates image intensity and camera metadata to different noise distributions. To this end, the baseline network needs to be extended to separate the different noise contributions and to process image and metadata together. Moreover, we expect an improved noise estimation accuracy as a result of learned awareness of separate noise sources and thus increased physical consistency.

In the following, we describe the three major extensions to the previous method for noise source estimation (right part of **Figure 2**): noise type identification (with or without the inclusion of camera metadata) and quantification of unexpected noise.



**Figure 2.** Noise level estimator versus proposed noise source and level estimator. Left: customized baseline estimator  $\text{DRNE}_{\text{cust.}}$ , which predicts the noise level of the input image’s total noise. Right: proposed noise source estimator that additionally employs camera metadata and predicts the noise levels of four different noise types. Architectural changes from the baseline are highlighted.

### 3.2.1. Noise Type Identification/Separation

In a first step, we duplicate the FCB and its preceding global max pooling layer to get three independent network branches. Each branch will predict the noise level of one noise type.

### 3.2.2. Inclusion of Camera Metadata

In a second step, we separate the camera’s metadata pertaining to the noise model into fixed and variable metadata (see **Table 1**). We assume the fixed metadata to be constant at training and inference times due to multiple reasons: 1) only parameters that in a sensitivity analysis lead to significant noise changes in the noise model are picked as variable parameters (see supp. material). From these parameters, we also fix 2) the offset, for simplicity, and the ones that iii) we consider as too difficult to obtain from a consumer-grade camera.

For the variable metadata, we survey existing camera systems in the literature to determine parameter ranges that are typical for our application scenarios (excluding unique systems for specialized use cases). The variable parameters are arranged into “minimal” and “full” metadata. We consider minimal metadata as easy to obtain (camera gain [digital gain for simplicity] and exposure time are typically configurable, while most camera systems comprise a temperature sensor to approach dark current compensation) and full metadata as more comprehensively include parameters often provided by the camera manufacturer. For comparison, we derive three models, where each one is fed with different metadata: one without any (w/o-Meta), one with minimal (Min-Meta), and one with full metadata (Full-Meta).

In preparation to use the metadata as input for the neural network, each parameter is first normalized to a floating point number in the range [0,1] (using the physical units, and the respective minimum and maximum values in Table 1). Subsequently, all

**Table 1.** Camera metadata used for noise source estimation. We split these into fixed and variable parameters, and consider only variable ones. Fixed parameters and all parameter definitions can be found in Supporting Information.

Variable parameter	Value range
Minimal metadata	
Camera gain	[0,24] dB
Exposure time	[0.001, 0.2] s
Sensor temperature	[0,80] °C
Full metadata	
Dark signal figure of merit (FoM)	[0,1]
Full well capacity	[2,100] × 10 <sup>3</sup> e <sup>-</sup>
Pixel clock rate	[8,150] × 10 <sup>6</sup> Hz
Sense node gain	[1,5] × 10 <sup>-6</sup> mm
Sense node	-
Reset factor	[0,1]
Sensor pixel size	[0.0009, 0.01] mm
Sensor type	{CCD, CMOS}
Thermal white noise	[1,60] × 10 <sup>-9</sup> Hz

parameters are combined and passed as a single array to the neural network (see attached source code for details). Inside the network, the metadata subset associated to its respective noise type is then concatenated with the output of the corresponding global max pooling layer and passed into its FCB. Note that using FCBs over the noise model itself to estimate the noise levels is 1) fast (using a graphics processing unit (GPU)), 2) allows us to train on real noise data that is not covered by the noise model, and 3) allows us to perform non-trivial feature-wise fusion with the feature maps from the processed input image.

### 3.2.3. Unexpected Noise Quantification

In the proposed system (Figure 2, right), we add a fourth FCB that quantifies unexpected noise, i.e., when the metadata does not agree with the considered image noise model. If we ensure that image noise is only generated inside the camera system (by preventing image pre- and post-processing) and assume a radiometrically calibrated camera (including a correct determination of the relevant metadata), there are two reasons for noise-metadata mismatch: 1) corrupted metadata (e.g., by camera malfunctioning) or 2) unmodeled noise sources (e.g., also by hardware damages, or a general mismatch between the noise model and the real image noise).

Specifically, we train this fourth FCB to quantify

$$\xi_{M/I} \doteq \sigma_{\text{Model}} - \sigma_{\text{Image}} \stackrel{(1)}{=} \frac{\sqrt{\sigma_{\text{PN}}^2(M_1) + \sigma_{\text{DCSN}}^2(M_2) + \sigma_{\text{RN}}^2(M_3)}}{\sqrt{\sigma_{\text{PN}}^2(M'_1) + \sigma_{\text{DCSN}}^2(M'_2) + \sigma_{\text{RN}}^2(M'_3)}} \quad (2)$$

with  $\xi_{M/I}$  normalized to [-1,1] for training, the total image noise  $\sigma_{\text{Image}}$ , the total modeled noise  $\sigma_{\text{Model}}$ , and metadata sets  $M_1, \dots, M_3$ , and altered sets  $M'_1, \dots, M'_3$  having a different randomly generated camera gain. The metadata sets  $M_{(\cdot)}$  are only fed to the FCBs (corresponding to noise level  $\sigma_{\text{Model}}$ ) while the altered sets  $M'_{(\cdot)}$  are used to corrupt the image (with corresponding noise level  $\sigma_{\text{Image}}$ ). In this way, the network learns to capture the mismatch between the metadata and the image noise in  $\xi_{M/I}$ .

With all the aforementioned extensions, the number of network parameters slightly increases, from 336 to 345 k.

## 3.3. Training Details

We utilize an almost noise-free dataset with natural images (TAMPERE21<sup>[36]</sup>), whose noise variance is ensured to be  $\sigma^2 < 1$ . These images are first augmented by a small random image intensity change of [-20,20] DN and afterward corrupted with noise generated by the noise model of ref. [19]. Each image patch is corrupted independently with its own set of randomly generated variable metadata. In this way, we generate ≈103 k data tuples to train the estimators in a supervised manner. Our motivation to train on simulated noise only is to cover a large extent of different metadata and to keep the limited real noise data available for model evaluation. The network’s branches are collectively trained utilizing the mean squared error loss function along with the Adam optimizer<sup>[37]</sup> and an initial



learning rate of  $10^{-4}$ . Further implementation details and the training configuration can be found in the code base.

## 4. Experimental Section

We first described the datasets used and the image noise applied (Section 4.1). Depending on whether a dataset includes ground truth (GT) labels or not, we conducted either quantitative or qualitative experiments. Our quantitative experiments comprised performance evaluations on simulated and real-world data (Section 4.2). In qualitative experiments, we evaluated our methods in real-field campaigns and on three use cases of unexpected noise (Section 4.3). In addition to the ability to quantify individual noise sources, we additionally demonstrated the improved total noise estimation performance on the downstream task of real-world image denoising (Section 4.4). Subsequently, we analyzed the effects of each camera metadata on noise estimation in comparison to the applied theoretical noise model (Section 4.5). Finally, we provided runtime measurements (Section 4.6).

We compared our proposed estimators against 1)  $B + F$ ,<sup>[10]</sup> DRNE<sub>cust.</sub>, principal component analysis (PCA)<sup>[4]</sup> and Poisson-Gaussian estimation-net (PGE-Net)<sup>[15]</sup> in the case of  $\sigma_{\text{Total}}$ , 2) PGE-Net for  $\sigma_{\text{PN}}$ , and 3) noise model predictions from the respective metadata for all individual noise levels  $\sigma_{i \in \{\text{PN,DCSN,RN}\}}$ . Note that PGE-Net was only applicable in the quantitative experiments, since it required (unnoised) GT images to calculate  $\hat{\sigma}_{i \in \{\text{PN,Total}\}}$ .

All experiments were executed on an Intel Xeon W-2145 central processing unit and an NVIDIA Quadro Ray tracing texel eXtreme 6000 GPU, with the neural networks running on the GPU. Noise levels were reported as digital numbers in the range  $[0, 255]_{\text{DN}}$ .

### 4.1. Datasets

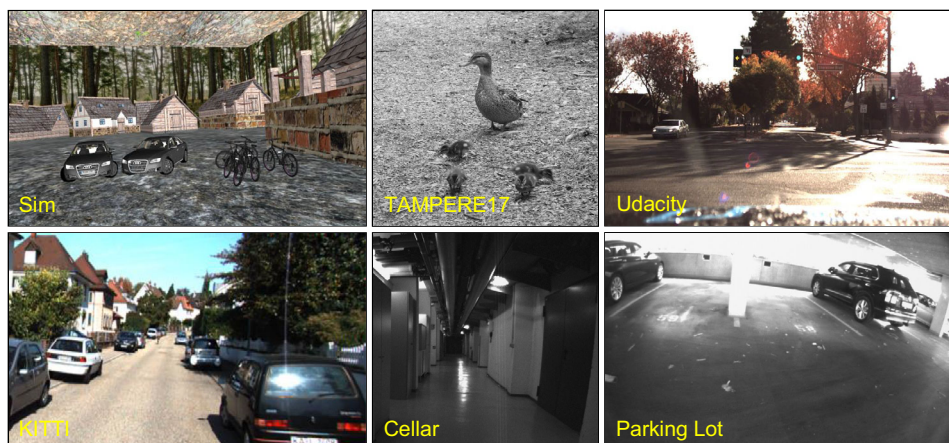
We augmented four datasets with GT labels and two datasets with pseudo GT labels (Figure 3).

#### 4.1.1. Datasets with GT

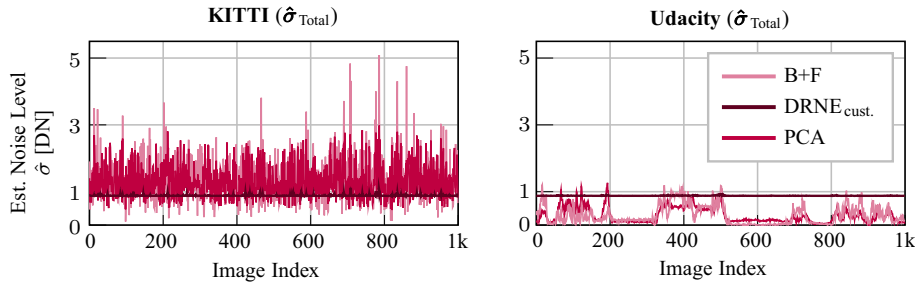
We employed one simulated and three real-world datasets: Sim, KITTI,<sup>[38]</sup> TAMPERE17,<sup>[39]</sup> and Udacity.<sup>[40]</sup> Sim was created with the simulator<sup>[41]</sup> to provide accurate GT for noise estimation. It comprised 1000 images of a village environment acquired from different viewpoints and included vehicles, such as cars and bikes. Similar to our training dataset TAMPERE21 (Section 3.3), TAMPERE17 provided 300 natural images with a controlled noise level of  $\sigma^2 < 1$ . From TAMPERE17, we used the grayscale version. KITTI and Udacity contained images from transportation scenarios. From KITTI, we used the annotated object detection sub-dataset and from Udacity sub-dataset #2. We considered only the first 1000 images from both datasets to match the number with Sim and reduce computation time. Depending on the respective image size, one image yielded several image patches.

Note that KITTI and Udacity did not include noise control. To assess how much noise both original datasets already contained, we applied three state-of-the-art noise level estimators (cf. Section 2). The results in Figure 4 indicated similarly small noise levels for Udacity as for both TAMPERE datasets, but significantly higher noise for KITTI. For this reason, we considered Udacity in the main article and provided KITTI results in Supporting Information.

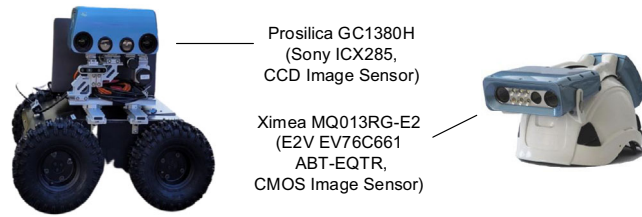
We corrupted all datasets with simulated or real-world noise. In the simulated case, we added noise to the images like our training dataset (cf. Section 3.3). In the real-world case, we generated in total 12 k RN and DCSN image tuples ( $I_{\text{RN}}, I_{\text{DCSN}}$ ) with about 600 different metadata sets from two different camera systems (we investigated several more camera types, e.g., Realsense D435i RGB and Huawei P30, but we reached the point where camera manufacturers would only provide metadata for private usage (i.e., not for publication) behind a non-disclosure agreement. Thus, we could not include them in this article) that we abbreviated according to their implemented camera sensors: ICX285<sup>[42]</sup> and EV76C661<sup>[43]</sup> (Figure 5). The first one was considered a scientific-grade charge-coupled device (CCD) and the latter was an industrial-grade CMOS camera system. PN was calculated synthetically as the quantum nature of light determines



**Figure 3.** Datasets. Exemplary image snippets from Sim ( $896 \times 768$  px), TAMPERE17 ( $512 \times 512$  px), Udacity ( $1920 \times 1200$  px), KITTI ( $1242 \times 375$  px), Cellar, and Parking Lot (ICX285:  $1360 \times 1024$  px, EV76C661:  $1280 \times 1024$  px).



**Figure 4.** Noise estimation of uncorrupted KITTI and Udacity datasets. The reference methods estimate significant noise in KITTI images ( $\hat{\sigma} \leq 5$ ) and low noise in Udacity data ( $\hat{\sigma} \leq 1.25$ ).



**Figure 5.** Camera systems. ICX285 is attached on an autonomous robotic platform and EV76C661 on an inspection helmet.

PN to strictly follow the Poisson distribution. Details about the real-world noise acquisition, noise post-processings, and used metadata could be found in Supporting Information.

#### 4.1.2. Datasets without GT

We collected two datasets from field campaigns without GT labels: Cellar and Parking Lot. Both datasets contained about 1000 grayscale images from respective eponymous environments and were recorded with both camera systems. We ensured high noise levels by applying the minimum exposure time of 1 ms (to capture low but detectable signals), maximum gain of 24 dB (to strongly amplify signal and noise without saturation), and by disabling all image post-processing (that could reduce noise).

We further evaluated the fourth noise type  $\xi_{M/I}$  as part of these field campaign experiments to demonstrate the detection of unexpected noise during operation time. Therefore, we split these experiments into two cases:  $\xi_{M/I} = 0$  and  $\xi_{M/I} \neq 0$ . The case  $\xi_{M/I} \neq 0$  was further subdivided into  $\xi_{M/I} < 0$  and  $\xi_{M/I} > 0$ . For  $\xi_{M/I} < 0$ , we simulated an additional image noise source by adding randomly generated Gaussian noise  $\mathcal{N}(\mu = 0, \sigma = 5\text{DN})$  to the images. For  $\xi_{M/I} > 0$ , we increased the model noise by synthetically doubling the value of the camera metadata thermal white noise. This parameter adjustment could be interpreted as a mis-calibration of the camera sensor's readout profile or a malfunctioned sensor component (e.g., the source follower). Moreover, we demonstrated the case of doubling the metadata sensor temperature in Supporting Information.

## 4.2. Quantitative Experiments

Metrics. We followed<sup>[44]</sup> and evaluated our noise source estimators in terms of accuracy  $\text{Bias} \doteq |\mathbb{E}[\hat{\sigma} - \sigma]|$ , robustness

$\text{Std} \doteq (\mathbb{E}[(\hat{\sigma} - \mathbb{E}(\hat{\sigma}))^2])^{1/2}$ , and overall performance  $\text{RMS} \doteq (\text{Bias}^2(\hat{\sigma}) + \text{Std}^2(\hat{\sigma}))^{1/2}$ , where  $\hat{\sigma}$  is the estimated noise level and  $\sigma$  is the true noise level. Smaller Root Mean Square (RMS), Bias, and Std values indicate better performance.

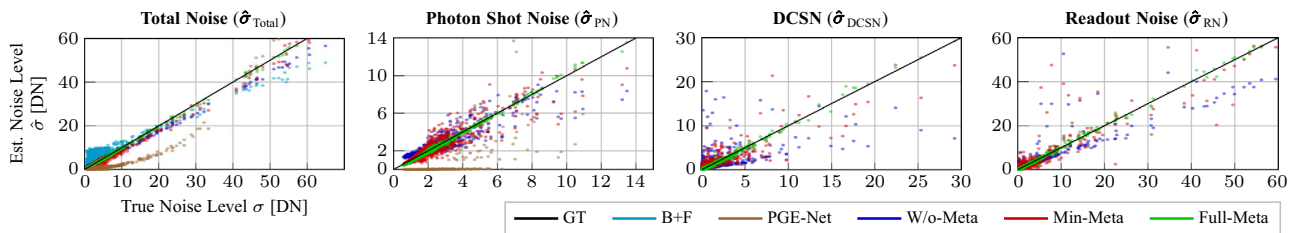
#### 4.2.1. Simulated Noise

The performance on the synthetically added noise datasets are summarized in **Table 2**, while mean noise estimation results on Sim are depicted in **Figure 6**.

Let us focus on results from Table 2 first. Among the reference methods, we observed that PGE-Net performed worst due to underestimation (cf. Figure 6), which agrees with ref. [15]. We could further see that  $\text{DRNE}_{\text{cust.}}$  generally produced better results than B + F. This observation matched ref. [35]. Considering our proposed methods, we observed that all three estimators accurately and robustly determine  $\sigma_{\text{Total}}$ , where Full-Meta was generally best, and Full-Meta and w/o-Meta performed slightly more robust than Min-Meta (smaller Std). In comparison to the reference methods, Full-Meta was on par with  $\text{DRNE}_{\text{cust.}}$ . When it came to noise source estimation, Full-Meta performed best. Both accuracy and robustness span sub-intensity levels for all three noise sources in all three datasets. w/o-Meta and Min-Meta also accurately quantified the single-noise types within sub-intensity levels on average (small bias). However, they had worse robustness in all datasets, particularly for DCSN and RN (large Std). We considered that this might be a problem of insufficient model capacity, but increasing the number of layers and neurons of the FCBs did not produce any change. We further made two detailed observations: all three methods estimated PN best, and Min-Meta determined the DCSN amount more robustly than w/o-Meta. We attributed the former observation to the strong link between image intensity and PN in the noise model, and the weaker influence of any metadata. However, only Full-Meta obtained the camera's full well capacity parameter, which seemed to slightly improve PN estimation. The more robust DCSN estimation performance of Min-Meta could be ascribed to its access to temperature and exposure time metadata, since both had a major impact on thermal noise.<sup>[19]</sup> The significance of metadata on separating the noise sources was further underpinned by the minor performance on RN estimation (as the minimal metadata only had a minor impact on the noise model) and by the prevailing performance

**Table 2.** Noise source estimation on synthetically corrupted datasets. The simulated noise is generated on the basis of randomly simulated camera sensors. The best results per method and dataset are highlighted in bold.

		Photon shot noise			DCSN			Readout noise			Total noise		
		Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS
Sim	B + F <sup>[10]</sup>	-	-	-	-	-	-	-	-	-	2.51	3.00	3.91
	DRNE <sub>cust.</sub>	-	-	-	-	-	-	-	-	-	<b>0.07</b>	<b>0.23</b>	<b>0.23</b>
	PCA <sup>[4]</sup>	-	-	-	-	-	-	-	-	-	0.75	1.07	1.30
	PGE-Net <sup>[15]</sup>	1.74	3.02	3.49	-	-	-	-	-	-	3.23	4.36	5.43
	W/o-Meta	<b>0.01</b>	0.75	0.75	0.35	4.23	4.24	0.35	3.40	3.42	0.50	1.22	1.32
	Min-Meta	0.05	0.75	0.76	0.13	2.82	2.83	0.13	3.38	3.39	0.47	0.97	1.08
	Full-Meta	0.09	<b>0.07</b>	<b>0.09</b>	<b>0.07</b>	<b>0.34</b>	<b>0.35</b>	<b>0.09</b>	<b>0.46</b>	<b>0.47</b>	0.16	0.29	0.33
Tamp.17	B + F <sup>[10]</sup>	-	-	-	-	-	-	-	-	-	2.22	4.19	4.74
	DRNE <sub>cust.</sub>	-	-	-	-	-	-	-	-	-	0.21	0.44	0.49
	PCA <sup>[4]</sup>	-	-	-	-	-	-	-	-	-	2.81	3.04	4.14
	PGE-Net	2.06	1.72	2.68	-	-	-	-	-	-	3.15	3.34	4.59
	W/o-Meta	0.16	0.84	0.85	<b>0.07</b>	3.11	3.11	<b>0.02</b>	3.04	3.04	<b>0.02</b>	1.18	1.18
	Min-Meta	<b>0.09</b>	0.82	0.83	0.21	2.01	2.02	0.39	3.73	3.75	<b>0.02</b>	1.05	1.05
	Full-Meta	0.10	<b>0.13</b>	<b>0.16</b>	0.09	<b>0.29</b>	<b>0.30</b>	0.17	<b>0.37</b>	<b>0.41</b>	0.05	<b>0.43</b>	<b>0.43</b>
Udacity	B + F <sup>[10]</sup>	-	-	-	-	-	-	-	-	-	1.09	2.19	2.44
	DRNE <sub>cust.</sub>	-	-	-	-	-	-	-	-	-	0.24	0.50	0.54
	PCA <sup>[4]</sup>	-	-	-	-	-	-	-	-	-	0.70	0.93	1.17
	PGE-Net	1.58	2.05	2.59	-	-	-	-	-	-	3.04	3.70	4.79
	W/o-Meta	<b>0.05</b>	0.54	0.54	0.28	3.31	3.33	0.45	2.54	2.58	0.44	1.39	1.46
	Min-Meta	0.19	0.66	0.68	<b>0.03</b>	2.21	2.21	0.27	2.38	2.40	<b>0.11</b>	0.88	0.89
	Full-Meta	0.06	<b>0.14</b>	<b>0.15</b>	0.04	<b>0.30</b>	<b>0.30</b>	<b>0.10</b>	<b>0.44</b>	<b>0.45</b>	0.14	<b>0.42</b>	<b>0.45</b>



**Figure 6.** Noise source estimation on synthetic noise (dataset: Sim, camera: random). Each dot represents the mean noise estimation of one image. The plots of DRNE<sub>cust.</sub> and PCA are omitted in the case of  $\hat{\sigma}_{\text{Total}}$  due to a strong similarity with the other plots (to avoid clutter).

of Full-Meta, which had access to the largest amount of metadata.

Figure 6 confirms the results of Table 2. It further indicates the increasing bias for w/o-Meta, the increasing Std (spread of the distributions) for w/o-Meta and Min-Meta with increasing noise levels  $\sigma_{i \in \{\text{Total, PN, DCSN, RN}\}}$ .

In summary, only Full-Meta with access to the full set of camera metadata could accurately and robustly quantify the contribution of each noise source. Although all variants of the proposed method could estimate the total noise level well, the lack of camera metadata for w/o-Meta and Min-Meta made it difficult to disambiguate the origin of the noise (i.e., identify the noise sources).

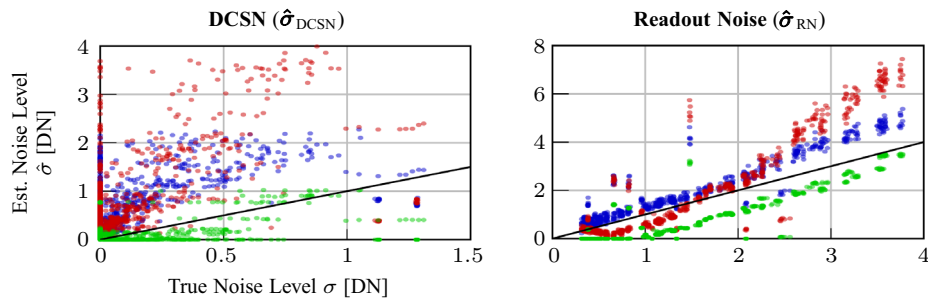
#### 4.2.2. Real-World Noise

Next we discussed the estimation performances of the real-world DCSN/RN produced by ICX285 and EV76C661 using Table 3 and Figure 7. Note that these two noise-optimized sensors produced lower noise levels compared to our simulated sensors ( $\sigma_{i \in \{\text{DCSN, RN}\}} \leq 5\text{DN}$ ). Both sensors lead to similar results; hence, we focused on ICX285 here and considered EV76C661 in Supporting Information.

In contrast to the fully simulated noise experiments (Table 2), the absolute DCSN and RN estimation performances of w/o-Meta and Min-Meta seem to have improved in Table 3. These results should not be overrated due to the generally

**Table 3.** Noise source estimation on real-world noise extracted from a Sony ICX285 CCD Sensor. DCSN and RN with corresponding metadata were recorded from the camera. PN was generated synthetically using the real metadata. The best results per method and dataset are highlighted in bold.

		Photon shot noise			DCSN			Readout noise			Total noise		
		Bias	Std.	RMS	Bias	Std.	RMS	Bias	Std.	RMS	Bias	Std.	RMS
Sim	B + F <sup>[10]</sup>	–	–	–	–	–	–	–	–	–	3.12	1.60	3.51
	DRNE <sub>cust.</sub>	–	–	–	–	–	–	–	–	–	0.17	0.28	0.33
	PCA <sup>[4]</sup>	–	–	–	–	–	–	–	–	–	1.11	0.82	1.38
	PGE-Net <sup>[15]</sup>	3.01	1.22	3.25	–	–	–	–	–	–	3.11	1.26	3.35
	W/o-Meta	0.63	0.63	0.89	0.68	0.59	0.90	0.43	<b>0.61</b>	<b>0.75</b>	0.08	0.27	0.29
	Min-Meta	1.03	0.21	1.05	0.80	0.86	1.18	<b>0.26</b>	1.35	1.38	0.77	0.65	1.00
	Full-Meta	<b>0.14</b>	<b>0.09</b>	<b>0.17</b>	<b>0.15</b>	<b>0.45</b>	<b>0.47</b>	0.82	0.95	1.25	<b>0.04</b>	<b>0.19</b>	<b>0.20</b>
Tamp.17	B + F <sup>[10]</sup>	–	–	–	–	–	–	–	–	–	2.71	3.54	4.43
	DRNE <sub>cust.</sub>	–	–	–	–	–	–	–	–	–	0.37	0.40	0.55
	PCA <sup>[4]</sup>	–	–	–	–	–	–	–	–	–	3.07	2.77	4.14
	PGE-Net <sup>[15]</sup>	3.03	1.35	3.32	–	–	–	–	–	–	2.74	1.71	3.23
	W/o-Meta	0.46	0.68	0.82	0.83	0.55	1.00	0.74	<b>0.76</b>	<b>1.06</b>	0.26	0.53	0.59
	Min-Meta	0.95	0.28	0.99	0.85	0.82	1.18	<b>0.37</b>	1.36	1.41	0.59	0.78	0.98
	Full-Meta	<b>0.22</b>	<b>0.14</b>	<b>0.26</b>	<b>0.14</b>	<b>0.41</b>	<b>0.44</b>	0.85	0.87	1.21	<b>0.13</b>	<b>0.36</b>	<b>0.38</b>
Udacity	B + F <sup>[10]</sup>	–	–	–	–	–	–	–	–	–	0.33	0.58	0.66
	DRNE <sub>cust.</sub>	–	–	–	–	–	–	–	–	–	<b>0.01</b>	0.53	0.53
	PCA <sup>[4]</sup>	–	–	–	–	–	–	–	–	–	0.14	0.63	0.64
	PGE-Net <sup>[15]</sup>	2.44	1.02	2.64	–	–	–	–	–	–	3.00	1.48	3.35
	W/o-Meta	0.44	0.49	0.66	0.64	0.57	0.85	<b>0.27</b>	<b>0.65</b>	<b>0.70</b>	0.04	<b>0.27</b>	<b>0.27</b>
	Min-Meta	0.63	0.21	0.66	0.76	0.84	1.14	0.28	1.33	1.36	0.41	0.68	0.79
	Full-Meta	<b>0.04</b>	<b>0.10</b>	<b>0.11</b>	<b>0.17</b>	<b>0.44</b>	<b>0.47</b>	0.87	0.97	1.30	0.25	0.30	0.39



**Figure 7.** Noise source estimation on real-world noise (dataset: Sim, camera: ICX285). Compare to Figure 6.

smaller noise levels and because the errors in the fully simulated cases started to majorly increase for noise levels  $\sigma_{i \in \{DCSN, RN\}} \geq 5\text{DN}$ . However, we observed two significant relative performance changes: Full-Meta worsened for RN and w/o-Meta improved for DCSN/RN. We attributed the change of both methods in the case of RN to the simulation-reality gap of the noise model that w/o-Meta coincidentally profits from (cf. Figure 7), because both methods were trained on simulated data only where it was shown that Full-Meta matched it better (Table 2). In the case of DCSN, the better performance of w/o-Meta was misleading, since only Full-Meta seemed to approximately fit the GT, while the others failed (see Figure 7).

These errors also propagated to the overall noise estimation. The estimations of the simulated PN did not change significantly.

In summary, despite the simulation-to-reality gap observed in these experiments, the access to the full metadata still led to the best results in terms of noise source quantification, thus providing evidence for the generalization capabilities of the method.

### 4.3. Experiments on Real-world Platforms

We recorded datasets Cellar and Parking Lot with camera systems ICX285 and EV76C661 in field campaigns (Figure 5). For comparison, we used B + F, DRNE<sub>cust.</sub>, and PCA in the case



of total noise and the noise model predictions with live recorded metadata for the individual noise sources. Since we observed similar results for both cameras and both datasets, we focused on ICX285 and Cellar here, and considered the rest in Supporting Information. We first evaluated the raw dataset (Section 4.3.1) and subsequently tested three altered versions with unexpected noise (Section 4.3.2).

#### 4.3.1. Expected Noise ( $\sigma_{\text{Model}} \approx \sigma_{\text{Image}}$ )

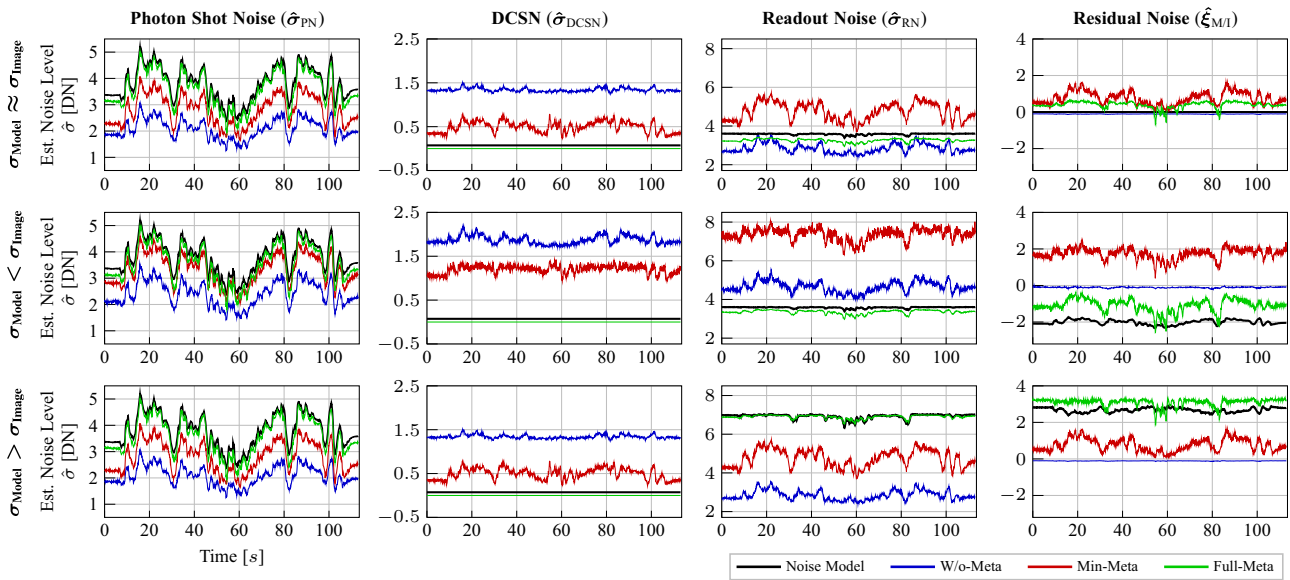
Let us first focus on the noise source identification (top row in **Figure 8**). We see that Full-Meta matches the noise model best with  $|\hat{\sigma}_{\text{Full-Meta}} - \hat{\sigma}_{\text{Reference}}| < 1\text{DN}$  in each noise case, followed by Min-Meta and w/o-Meta. These results were generally in accordance to the simulated noise evaluations in Section 4.2.1. The only significant difference we observed was that Min-Meta matched the relative value range of the PN noise model curve better than w/o-Meta (i.e., smaller Std). This could be explained with the camera gain parameter that Min-Meta obtained as one key parameter in the noise model to determine

PN (already is indicated on the simulated ICX285 in Table 3). The residual noise plot depicted only a small mismatch between the noise model and the detected image noise for Full-Meta and Min-Meta. Only the nearly constant value of w/o-Meta indicated that it did not learn to detect any residual noises. From this residual noise estimation of Full-Meta (and later results from Figure 8), we assumed for Cellar that

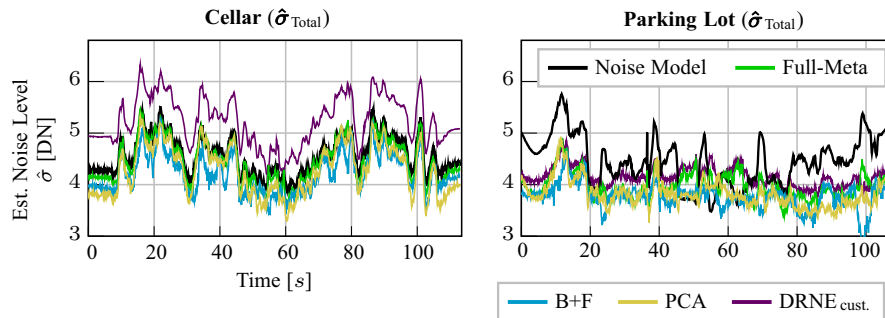
$$\xi_{M/1} \approx 0 \Rightarrow \sigma_{\text{Model}} \approx \sigma_{\text{Image}} \quad (3)$$

In the total noise inspection (**Figure 9**), we considered only Full-Meta from our proposed methods to avoid clutter. We saw from both plots that Full-Meta produced similar estimations as the reference methods. Hence, we considered its results as plausible.

In summary, the results agreed with those of the synthetic noise experiments, meaning that our model was applicable to an actual real-world robotic platform. The more metadata were available, the better the noise source estimation of all noise types (with Full-Meta as the best method).



**Figure 8.** Noise source estimation with and without unexpected noise (dataset: Cellar, camera: ICX285). Top row: estimation on the uncorrupted dataset. Middle row: image noise increased by random Gaussian noise  $\mathcal{N}(\mu = 0, \sigma = 5 \text{ DN})$ . Bottom row: model noise increased by doubling camera parameter thermal white noise.



**Figure 9.** Total noise estimation (datasets: Cellar and Parking Lot, camera: ICX285). Compare the left plot to Figure 8.

### 4.3.2. Unexpected Noise ( $\sigma_{\text{Model}} \neq \sigma_{\text{Image}}$ )

Here, we evaluated three scenarios where we synthetically increased image noise or model noise to reach  $\xi_{M/I} < 0$  (i.e.,  $\sigma_{\text{Model}} < \sigma_{\text{Image}}$ ) or  $\xi_{M/I} > 0$  (i.e.,  $\sigma_{\text{Model}} > \sigma_{\text{Image}}$ ), respectively. We investigated these scenarios on the basis of the raw Cellar dataset for that we assumed that the applied noise model followed the actual image noise (i.e., Equation (3):  $\xi_{M/I} \approx 0$ ).

One scenario of the form  $\sigma_{\text{Model}} < \sigma_{\text{Image}}$ : in our first scenario, we increased the image noise by adding randomly sampled Gaussian noise from  $\mathcal{N}(\mu = 0, \sigma_{\mathcal{N}} = 5\text{DN})$  to the raw Cellar images. Note that this Gaussian noise was statistically independent from the other image noise sources and so its noise level was added in quadrature to the new total image noise level  $\sigma_{\text{Image}+\mathcal{N}}$  (cf. (1)). We calculated the resulting GT  $\xi_{M/I}$  as

$$\xi_{M/I} \stackrel{(2)}{=} \sigma_{\text{Model}} - \sigma_{\text{Image}+\mathcal{N}} \stackrel{(3)}{\approx} \sigma_{\text{Model}} - \sqrt{\sigma_{\text{Model}}^2 + \sigma_{\mathcal{N}}^2} \quad (4)$$

The middle row of Figure 8 illustrates the results. We expected only a reduction of  $\xi_{M/I}$  and unchanged values otherwise, with respect to the first row. It could be seen that only Full-Meta captured the unexpected noise (note the initial error of  $\approx 0.5$  DN was propagated), whereas w/o-Meta remained unchanged (cf. Section 4.3.1) and Min-Meta incorrectly estimate increased

values. Furthermore, w/o-Meta and Min-Meta split  $\sigma_{\mathcal{N}}$  among the other noise sources (especially Min-Meta increases  $\hat{\sigma}_{\text{RN}}$  significantly). Only Full-Meta maintained its noise source estimated values.

Two scenarios of the form  $\sigma_{\text{Model}} > \sigma_{\text{Image}}$ : in this second test, we increased the model noise by doubling the metadata thermal white noise. This parameter only affected Full-Meta. The new GT noise levels were calculated using the noise model. (in the third test, we prepared an example with a doubled metadata sensor temperature, however, without new findings; thus, it was treated in Supporting Information).

The results are shown in the bottom row of Figure 8. In this case, we expected an increasing  $\hat{\sigma}_{\text{RN}}$  in accordance to the increased thermal white noise, an increasing  $\xi_{M/I}$  (which indicated the unexpected higher model noise) and unchanged values otherwise. We could see that Full-Meta met these expectations (note the initial propagated error here as well).

We concluded that unexpected noise in either images or from metadata could only be reliably quantified with the full set of variable camera metadata.

### 4.4. Experiments on Real-World Image Denoising

Our proposed noise source estimator was able to improve total image noise estimation (see Table 2 and 3), thereby offering

**Table 4.** Denoising performance for real-world images (camera: ICX 285). Best PSNR (dB  $\uparrow$ ) and SSIM ( $\uparrow$ ) scores per dataset, noise level, and metric are highlighted in bold, the second best are underlined (in the case of equal numbers, the decision is made on the basis of further decimal places).

Method		Number of raw images for averaging				
		1	2	4	8	16
Cellar	Raw	34.75/0.7730	37.61/0.8703	40.42/0.9322	43.21/0.9680	46.13/0.9872
	DRNE <sub>cust.</sub> + BM3D <sup>[45]</sup>	43.01/0.9803	44.47/0.9853	45.53/0.9886	46.49/0.9913	<u>47.59/0.9932</u>
	w/o-Meta + BM3D <sup>[45]</sup>	41.42/0.9671	43.83/0.9817	45.01/0.9861	46.26/0.9905	47.56/0.9930
	Min-Meta + BM3D <sup>[45]</sup>	<u>43.30/0.9818</u>	<u>44.72/0.9864</u>	<u>45.67/0.9899</u>	<u>46.49/0.9912</u>	47.32/0.9922
	Full-Meta + BM3D <sup>[45]</sup>	<b>43.74/0.9839</b>	<b>45.00/0.9875</b>	<b>45.99/0.9901</b>	<b>46.68/0.9916</b>	<b>47.63/0.9934</b>
	DRNE <sub>cust.</sub> + NLM <sup>[46]</sup>	42.46/0.9793	43.91/0.9841	45.05/0.9874	46.09/0.9902	47.32/0.9925
	w/o-Meta + NLM <sup>[46]</sup>	41.08/0.9701	43.33/0.9812	44.66/0.9322	45.94/0.9897	47.28/0.9924
	Min-Meta + NLM <sup>[46]</sup>	42.77/0.9809	44.18/0.9852	45.18/0.9879	46.09/0.9902	46.92/0.9911
	Full-Meta + NLM <sup>[46]</sup>	43.19/0.9828	44.47/0.9863	45.50/0.9889	46.26/0.9905	47.35/0.9927
	FBI-Denoiser <sup>[15]</sup>	41.69/0.9830	42.07/0.9851	42.29/0.9865	42.41/0.9871	42.62/0.9880
	Blind2Unblind <sup>[47]</sup>	43.02/0.9515	43.67/0.9574	44.10/0.9616	44.41/0.9643	44.77/0.9660
	Parking Lot	Raw	31.09/0.7890	32.34/0.8780	33.29/0.9330	34.07/0.9625
DRNE <sub>cust.</sub> + BM3D <sup>[45]</sup>		33.39/0.9546	33.78/0.9639	34.11/0.9713	<u>34.51/0.9770</u>	<u>35.39/0.9820</u>
w/o-Meta + BM3D <sup>[45]</sup>		33.04/0.9394	33.70/0.9612	34.07/0.9705	34.50/0.9770	35.37/0.9817
Min-Meta + BM3D <sup>[45]</sup>		<u>33.47/0.9573</u>	<u>33.83/0.9651</u>	<u>34.11/0.9710</u>	34.44/0.9749	35.20/0.9780
Full-Meta + BM3D <sup>[45]</sup>		33.40/0.9553	33.81/0.9648	<b>34.11/0.9715</b>	<b>34.51/0.9771</b>	<b>35.40/0.9823</b>
DRNE <sub>cust.</sub> + NLM <sup>[46]</sup>		33.14/0.9464	33.56/0.9567	33.92/0.9655	34.36/0.9731	35.29/0.9799
w/o-Meta + NLM <sup>[46]</sup>		32.93/0.7890	33.52/0.9559	33.91/0.9657	34.37/0.9738	35.26/0.9794
Min-Meta + NLM <sup>[46]</sup>		33.11/0.9447	33.52/0.9547	33.84/0.9622	34.17/0.9670	34.88/0.9709
Full-Meta + NLM <sup>[46]</sup>		33.14/0.9468	33.56/0.9566	33.92/0.9657	34.37/0.9734	35.31/0.9804
FBI-Denoiser <sup>[15]</sup>		32.52/0.9450	32.67/0.9522	32.76/0.9573	32.87/0.9604	33.05/0.9630
Blind2Unblind <sup>[47]</sup>		<b>33.61/0.9156</b>	<b>33.86/0.9282</b>	34.05/0.9379	34.27/0.9441	34.74/0.9488

potential advantages for downstream vision tasks. Although we used the symptom-fighting denoising as a rationale for noise source estimation, denoising was the most studied downstream application for noise level estimation and therefore best suited to assess the effects of estimating total noise more accurately.

We investigated the effect of more accurate total noise level estimation on denoising on the example of two traditional denoisers that input expected noise levels (BM3D<sup>[45]</sup> and non-local means (NLM)<sup>[46]</sup>) and compare results to two state-of-the-art learning-based denoisers (FBI-Denoiser<sup>[15]</sup> and Blind2Unblind<sup>[47]</sup>). BM3D and NLM both assumed Gaussian noise, the FBI denoiser internally used PGE-Net for Poisson-Gaussian noise estimation, and Blind2Unblind did not explicitly assume a noise distribution. We applied all denoisers with default parameter values and pre-trained weights provided by the respective authors (we selected respective weights for real-noise images that led to the best results for our datasets, i.e., “DND” weights for FBI-Denoiser and “raw RGB” weights for Blind2Unblind). For a fair comparison, all denoisers were applied to whole images. Denoising results are compared using peak signal-to-noise ratio (PSNR [dB]) and structural similarity index measure (SSIM).<sup>[48]</sup>

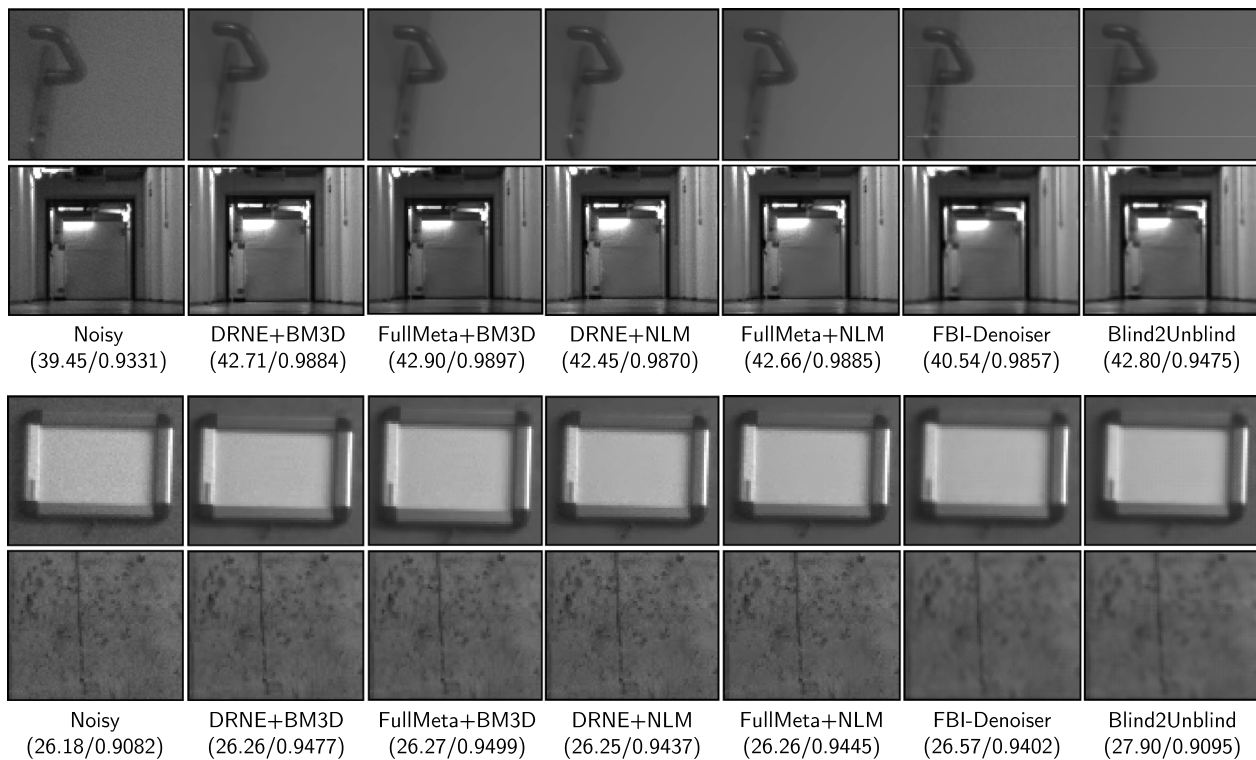
**Table 4** presents quantitative results using the ICX285 camera. For the Cellar scene, Full-Meta + BM3D led to the best scores in all cases, followed by Min-Meta + BM3D for cases of higher

noise (less images used for averaging), and DRNE<sub>cust.</sub> + BM3D for lower noise cases (more images for averaging). We observed similar results in combination with the NLM denoiser. This is in accordance with the results from Table 2 that DRNE<sub>cust.</sub> and Full-Meta perform best and in accordance with Table 3 that Min-Meta was not far off, but it counteracted the nonintuitive results from Table 3 that w/o-Meta occasionally led to more accurate total noise estimations. The denoising resulted rather underpin the intuition that the more metadata available for the noise source estimators, the better the total noise estimation. The learning-based denoisers performed less accurate than BM3D, which differed from the results reported in their respective original studies, as these denoisers were neither trained on large and diverse real-world datasets (with the weights we employed) nor fine-tuned to our datasets. The performance gap between the traditional and the learning-based denoisers increased with decreasing noise level.

We note similar results for the Parking Lot dataset with the difference that Blind2Unblind and Min-Meta both score best in the two highest noise cases. However, the better performance of Min-Meta compared to Full-Meta might be specific to BM3D, as Full-Meta was relatively more accurate when combined with NLM. Experiments on the EV76C661 camera yielded comparable findings (**Table 5**).

**Table 5.** Denoising performance for real-world noised images (camera: EV76C661). Best PSNR (dB ↑) and SSIM (↑) scores per dataset, noise level, and metric are highlighted in bold, the second best are underlined. Compare to Table 4.

Method		Number of raw images for averaging				
		1	2	4	8	16
Cellar	Raw	28.59/0.6052	30.73/0.7447	32.40/0.8517	33.80/0.9227	<b>35.36/0.9651</b>
	DRNE <sub>cust.</sub> + BM3D <sup>[45]</sup>	<u>33.30/0.9136</u>	<b>33.84/0.9221</b>	34.06/0.9280	<u>34.30/0.9342</u>	<u>34.85/0.9399</u>
	w/o-Meta + BM3D <sup>[45]</sup>	33.29/0.9132	33.83/0.9221	<u>34.08/0.9298</u>	34.27/0.9322	34.72/0.9337
	Min-Meta + BM3D <sup>[45]</sup>	33.28/0.9125	33.75/0.9192	34.05/0.9268	34.29/0.9339	34.83/0.9387
	Full-Meta + BM3D <sup>[45]</sup>	<b>33.33/0.9151</b>	<u>33.83/0.9209</u>	<b>34.10/0.9317</b>	<b>34.37/0.9388</b>	34.83/0.9386
	DRNE <sub>cust.</sub> + NLM <sup>[46]</sup>	33.22/0.9132	33.73/0.9200	33.95/0.9248	34.19/0.9296	34.71/0.9341
	w/o-Meta + NLM <sup>[46]</sup>	33.22/0.9133	33.74/0.9207	33.97/0.9263	34.17/0.9283	34.62/0.9299
	Min-Meta + NLM <sup>[46]</sup>	33.22/0.9131	33.69/0.9177	33.93/0.9238	34.18/0.9294	34.70/0.9332
	Full-Meta + NLM <sup>[46]</sup>	33.22/0.9133	33.73/0.9219	34.00/0.9285	34.24/0.9327	34.70/0.9333
	FBI-Denoiser <sup>[15]</sup>	33.18/0.9127	33.65/0.9196	33.82/0.9231	33.99/0.9251	34.42/0.9264
Blind2Unblind <sup>[47]</sup>	33.14/0.8921	33.60/0.9008	33.79/0.9075	33.99/0.9122	34.34/0.9149	
Parking lot	Raw	31.06/0.6734	33.31/0.8050	35.40/0.8953	37.12/0.9499	<b>37.78/0.9806</b>
	DRNE <sub>cust.</sub> + BM3D <sup>[45]</sup>	<u>35.94/0.9342</u>	<u>36.50/0.9416</u>	37.10/0.9476	37.47/0.9537	37.33/0.9605
	w/o-Meta + BM3D <sup>[45]</sup>	35.42/0.9198	36.38/0.9407	37.16/0.9502	37.56/0.9566	37.46/0.9661
	Min-Meta + BM3D <sup>[45]</sup>	35.62/0.9262	36.11/0.9334	37.18/0.9519	37.57/0.9568	37.43/0.9649
	Full-Meta + BM3D <sup>[45]</sup>	35.85/0.9324	36.43/0.9415	<u>37.19/0.9524</u>	<b>37.82/0.9643</b>	<u>37.70/0.9757</u>
	DRNE <sub>cust.</sub> + NLM <sup>[46]</sup>	35.67/0.9312	36.24/0.9377	36.87/0.9437	37.27/0.9498	37.78/0.9560
	w/o-Meta + NLM <sup>[46]</sup>	35.52/0.9295	36.27/0.9404	36.96/0.9467	37.35/0.9522	37.30/0.9604
	Min-Meta + NLM <sup>[46]</sup>	35.60/0.9314	36.13/0.9376	36.99/0.9483	37.36/0.9524	37.27/0.9594
	Full-Meta + NLM <sup>[46]</sup>	35.67/0.9320	36.28/0.9403	37.00/0.9488	37.55/0.9581	37.52/0.9694
	FBI-Denoiser <sup>[15]</sup>	35.58/0.9287	36.02/0.9345	36.51/0.9389	36.70/0.9420	36.59/0.9446
Blind2Unblind <sup>[47]</sup>	<b>36.37/0.8109</b>	<b>36.86/0.8243</b>	<b>37.37/0.8358</b>	<u>37.65/0.8446</u>	37.51/0.8511	



**Figure 10.** Exemplary denoising results for real-world noisy images (four averaged images, top rows: Cellar, bottom rows: Parking Lot). Brightness and contrast are adapted for better visualization. FullMeta + BM3D best removes the noise while preserving image details. In contrast, FBI-Denoiser and Blind2Unblind remove noise best visually, but smooth the entire image (both) and introduce square artifacts (Blind2Unblind). NLM tends to retain noise at edges.

Figure 10 illustrates qualitative results on the example of both scenes recorded with the ICX285 camera and a medium noise level (four averaged images each). It could be seen that the FullMeta + BM3D combination was the best at visually removing noise while preserving image detail, closely followed by DRNE + BM3D (e.g., FullMeta+BM3D restored the edges of the shadows less pixelated in the first row of Figure 10). In contrast, FBI-Denoiser and Blind2Unblind visually removed noise the best, but smooth the entire image (both methods, see especially bottom rows in Figure 10) and introduced square artifacts (Blind2Unblind). NLM tended to generally retain noise at edges (e.g., around the door handle in the first row and around the silver frame in the third row of Figure 10).

In summary, Full-Meta in combination with the traditional BM3D denoiser led to the best denoising results in most cases. This supported previous findings that Full-Meta generally estimated total noise levels the best and thus that noise estimation can benefit from camera metadata.

#### 4.5. Metadata Sensitivity Analysis

In this section, we investigated the individual influence of camera metadata on total noise level estimations. Building upon previous results, we focused only on Full-Meta and compared it to the theoretical noise model.

We conducted a black box analysis by uniformly sampling different input parameter values from the respective parameter

ranges for both approaches and observing respective outputs (i.e., the total noise estimations). One input parameter was sampled at a time and other parameters were fixed to their respective maximum value (to aim for sufficiently high noise levels). Note that the only image feature the noise model depended on was the image intensity, while Full-Meta might have learned to employ more image features (e.g., image noise). As we focused on the influence of camera metadata only, we only input uncorrupted homogeneous images with uniform intensities.

In the case of Full-Meta, we further omitted the residual noise estimation  $\xi_{M/I}$  to calculate the total noise level, as a mismatch between image noise and camera metadata was expected. Finally, we compared the estimated noise levels of both models to quantify the impact of each metadata and whether Full-Meta had learned the theoretical model (we considered deviations in  $[1,2]DN$  as minor but worth noting and those larger than  $2DN$  as significant).

We first examined the effect of each parameter on the estimated noise level according to the theoretical noise model (top part of Table 6). For each row, the bigger the difference between estimated noise levels in the “Min” and “Max” columns, the more important the parameter. The table shows that the full well capacity is most important because it determines the photon shot noise in the noise model, followed by the camera gain that amplifies noise. The pixel clock rate, thermal white noise, and sense node reset factor, which all contributed to readout noise, had a negligible effect on the estimated total noise level. Note that



**Table 6.** Input–output sensitivity analysis of Full-Meta (bottom) compared to the noise model (top). Input: one parameter is sampled at a time while the rest are fixed to respective maximum values (to generate high noise levels) and the (uncorrupted) mean image intensity to 128 DN (to avoid saturation). Parameter value ranges are provided in Table 1 and concrete sampled values in the appendix. Output: estimated total noise level (table cells, in DN) per input parameter configuration. ★: The influence of the pixel clock rate highly depends on metadata that we fixed during the experiments, such as the correlated double sampling dominant time constant. ★★: Simulated CCD sensor (CMOS sensor otherwise).

		Uniform samples of parameter value ranges										
		Min	1	2	3	4	5	6	7	8	Max	
Noise Model	Mean image Intensity	5.54	9.87	9.94	10.0	10.1	10.1	10.2	10.2	10.3	6.1	
	Minimal metadata											
	Camera gain	0.82	1.86	2.88	3.92	4.94	5.96	6.99	8.02	9.02	10.1	
	Exposure time	4.02	5.06	5.93	6.67	7.35	7.96	8.53	9.08	9.60	10.1	
	Sensor temperature	3.69	3.76	3.86	4.03	4.31	4.77	5.49	6.56	8.05	10.1	
	Full metadata											
	Dark signal FoM	3.96	5.02	5.89	6.65	7.34	7.95	8.53	9.06	9.57	10.1	
	Full well capacity	118.4	69.6	41.4	28.6	21.8	17.6	14.8	12.8	11.3	10.1	
	PixelClockRate★	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33	3.33	
	Sense node (SN) gain	12.9	11.7	11.1	10.8	10.6	10.4	10.3	10.2	10.1	10.1	
	SN reset factor	9.56	9.56	9.58	9.59	9.67	9.71	9.76	9.85	9.95	10.1	
	Sensor pixel size	4.05	4.34	4.80	5.38	6.05	6.79	7.57	8.39	9.21	10.1	
	ThermalWh.Noise★★	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.1	10.1	
	Noise Source estimator (Full-Meta)	Mean image Intensity	7.05	9.57	9.87	9.98	10.1	10.1	10.1	9.95	9.56	8.52
		Minimal metadata										
Camera gain		0.90	1.70	2.56	3.53	4.15	5.11	6.61	7.84	8.86	10.1	
Exposure time		5.31	6.07	6.70	7.24	7.77	8.26	8.67	9.03	9.39	10.1	
Sensor temperature		4.34	4.49	4.71	4.98	5.34	5.85	6.36	7.23	8.48	10.1	
Full metadata												
Dark signal FoM		4.76	6.17	6.47	7.20	7.71	8.19	8.68	9.19	9.78	10.1	
Full well capacity		167.2	62.7	34.4	27.7	21.4	16.3	14.7	13.03	11.5	10.1	
PixelClockRate★		4.22	4.22	4.22	4.22	4.22	4.22	4.22	4.22	4.22	4.22	
Sense node (SN) gain		13.5	12.8	12.1	11.4	10.8	10.6	10.5	10.4	10.2	10.1	
SN reset factor		8.23	8.44	8.64	8.77	8.85	8.94	9.07	9.38	9.72	10.1	
Sensor pixel size		4.88	5.22	5.56	6.01	6.51	7.05	7.72	8.39	9.08	10.1	
ThermalWh.Noise★★		9.46	9.54	9.71	9.87	10.0	10.1	10.1	10.1	10.1	10.1	

these three parameters were only insignificant for the considered set of fixed parameters in our experiments (see Supporting Information). To illustrate, for instance, the impact of the pixel clock rate on the total noise level was closely tied to the correlated double sampling dominant time constant.<sup>[19]</sup>

Comparing the noise level of the noise model with the estimations from the Full-Meta model (bottom part in Table 6), Full-Meta had generally learned the relations between input parameters and the noise levels. Yet, there were specific cases where deviations could be observed (indicated by colored values). Let us consider the two severe model deviations first (red values). The first differences were the estimated noise levels for minimum and maximum mean image intensities. This corresponded to the reduced noise estimation accuracy that we observed for under- and overexposed images (see Supporting Information). The second deviation could be observed for full well capacities  $\leq 24$  k electrons. The corresponding noise levels were most

different from the others learned by Full-Meta (that range between about  $[0, 13]$  DN). The farther the noise values departed from this range, the larger the observed model deviation. This indicated that these noise levels were underrepresented in the training data. Minor deviations from the noise model (orange values) were limited to small respective parameter values, with the exception of the sensor temperature. However, we did not see a specific pattern in these deviations; they were mostly slightly above 1 DN.

In conclusion, the Full-Meta model learned to capture the theoretical camera metadata relations, with notable exceptions for low and high exposed images, and large noise levels resulting from camera full well capacities  $\leq 24$  k electrons. The full well capacity and the camera gain could be identified as the most significant camera metadata, while pixel clock rate, sense noise reset factor, and thermal white noise could be neglected.

#### 4.6. Computational Cost

The computation time was determined by averaging the noise estimation inference times for 13.5k Udacity image patches (i.e., 100 Udacity images). We repeated the measurements five times and took the average to eliminate the influence of background processes and caching. We measured the following average runtimes per image patch: 1.4 ms (w/o-Meta), 1.3 ms (Min-Meta), 1.3 ms (Full-Meta), 1.2 ms (DRNE<sub>cust.</sub>), 0.1 ms (PGE-Net), 9.8 ms (PCA), and 13.2 ms (B + F + F). Note that PGE-Net was faster because it processed a whole image at once, but it did not estimate as many noise sources nor was as accurate as the proposed method(s).

#### 5. Conclusion

We have proposed a noise source estimator that quantifies contributions of individual camera noise sources using an image with metadata to tackle noise at its root causes as opposed to tackling its symptoms. It is memory efficient and runs in real time, and its broad range of learned camera systems makes it directly applicable to many mobile agents. Comparing the three versions of the estimator, we validated the natural hypothesis that the more camera metadata is available and relevant, the better the noise source identification (Full-Meta). Moreover, the developed estimator (Full-Meta) promotes a reliable application by its ability to detect unexpected influences in image noise and the metadata. We have evaluated its functionality in extensive experiments including real-world noise from two camera systems, a self-simulated and three standard datasets, the application in two field campaigns with unexpected image noise and metadata changes, and in a sensitivity analysis of the input parameters. Lastly, the improved total noise estimation has been demonstrated in the context of the downstream vision application of image denoising. In the future, our method could be integrated into a machine's feedback loop to perform automatic countermeasures.<sup>[35]</sup> As our estimators already show notable experimental performance, we follow the baseline<sup>[6]</sup> and leave ablations for the future.

#### Acknowledgements

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2002/1 "Science of Intelligence" – project number 390523135.

#### Conflict of Interest

The authors declare no conflict of interest.

#### Data Availability Statement

Please check the Data Reporting Checklist attached.

#### Keywords

machine learnings, noise estimations, perceptions, sensor artificial intelligences, smart sensing systems (with image sensors)

- [1] N. El-Atab, R. B. Mishra, F. Al-Modaf, L. Joharji, A. A. Alsharif, H. Alamoudi, M. Diaz, N. Qaiser, M. M. Hussain, *Adv. Intell. Syst.* **2020**, 2, 2000128.
- [2] Z. Jin, Z. Zhang, G. X. Gu, *Adv. Intell. Syst.* **2020**, 2, 1900130.
- [3] Y. Zhang, P. Hang, C. Huang, C. Lv, *Adv. Intell. Syst.* **2022**, 4, 2100211.
- [4] G. Chen, F. Zhu, P. A. Heng, in *Int. Conf. Computer Vision (ICCV)*, Santiago, Chile **2015**, pp. 477–485.
- [5] V. Jain, S. Seung, in *Conf. Neural Information Processing Systems (NIPS)*, Vol. 21, Vancouver, Canada **2008**.
- [6] H. Tan, H. Xiao, S. Lai, Y. Liu, M. Zhang, *Computational Intelligence and Neuroscience*, Hindawi Limited, Hindawi, London, UK **2019**.
- [7] W. Wang, X. Chen, C. Yang, X. Li, X. Hu, T. Yue, in *Int. Conf. Computer Vision (ICCV)*, Seoul, Korea **2019**, pp. 4111–4119.
- [8] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang, *IEEE Trans. Image Process.* **2017**, 26, 3142.
- [9] J. Janesick, *Scientific Charge-Coupled Devices*, Vol. 83, SPIE Press, Bellingham, WA, USA **2001**.
- [10] D.-H. Shin, R.-H. Park, S. Yang, J.-H. Jung, *IEEE Trans. Consum. Electron.* **2005**, 51, 218.
- [11] M. Uss, B. Vozel, V. Lukin, S. Abramov, I. Baryshev, K. Chehdi, *EURASIP J. Adv. Signal Process.* **2011**, 2011, 1–12.
- [12] B. Corner, R. Narayanan, S. Reichenbach, *Int. J. Remote Sens.* **2003**, 24, 689.
- [13] A. De Stefano, P. R. White, W. B. Collis, *EURASIP J. Adv. Signal Process.* **2004**, 2004, 1–8.
- [14] S. Pyatykh, J. Hesser, L. Zheng, *IEEE Trans. Image Process.* **2012**, 22, 687.
- [15] J. Byun, S. Cha, T. Moon, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2021**, pp. 5768–5777.
- [16] Q. Lyu, M. Guo, Z. Pei, *Appl. Soft Comput.* **2020**, 95, 106478.
- [17] T. Plotz, S. Roth, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2017**, pp. 1586–1595.
- [18] G. E. Healey, R. Kondepudy, *IEEE Trans. Pattern Anal. Mach. Intell.* **1994**, 16, 267.
- [19] M. Konnik, J. Welsh, *arXiv preprint arXiv:1412.4031*, **2014**.
- [20] A. J. Blanksby, M. J. Loinaz, D. Inglis, B. D. Ackland, in *Int. Electron Devices Meeting. IEDM Technical Digest*, Washington, DC, USA **1997**, pp. 205–208.
- [21] A. Foi, M. Trimeche, V. Katkovnik, K. Egiazarian, *IEEE Trans. Image Process.* **2008**, 17, 1737.
- [22] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, W. T. Freeman, *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, 30, 299.
- [23] C. Sutour, C.-A. Deledalle, J.-F. Aujol, *SIAM J. Imag. Sci.* **2015**, 8, 2622.
- [24] J. Yang, Z. Gan, Z. Wu, C. Hou, *IEEE Trans. Image Process.* **2015**, 24, 1561.
- [25] Y. Matsushita, S. Lin, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2007**, pp. 1–8.
- [26] H. Yao, in *MATEC Web of Conf.*, Vol. 42, EDP Sciences, Les Ulis Cedex A, France **2016**, p. 06004.
- [27] H. Yao, M. Zou, C. Qin, X. Zhang, *IEEE Trans. Multimedia* **2021**, 24, 640.
- [28] K. Wei, Y. Fu, J. Yang, H. Huang, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2020**, pp. 2758–2767.
- [29] Y. Zhang, H. Qin, X. Wang, H. Li, in *Int. Conf. Computer Vision (ICCV)*, Montreal, QC, Canada **2021**, pp. 4593–4601.

- [30] E. Onzon, F. Mannan, F. Heide, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2021**, pp. 7710–7720.
- [31] J. Chen, J. Chen, H. Chao, M. Yang, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2018**, pp. 3155–3164.
- [32] A. Abdelhamed, M. A. Brubaker, M. S. Brown, in *Int. Conf. Computer Vision (ICCV)*, Seoul, Korea **2019**, pp. 3165–3173.
- [33] K.-C. Chang, R. Wang, H.-J. Lin, Y.-L. Liu, C.-P. Chen, Y.-L. Chang, H.-T. Chen, in *European Conf. Computer Vision (ECCV)*, Glasgow, UK **2020**, pp. 343–358.
- [34] W. Chen, B. Qi, X. Liu, H. Li, X. Hao, Y. Peng, *Adv. Intell. Syst.* **2022**, *4*, 2200149.
- [35] M. Wischow, G. Gallego, I. Ernst, A. Börner, in *IEEE Transactions on Intelligent Transportation Systems*, IEEE, Piscataway, NJ **2023**, pp. 1–14.
- [36] S. G. Bahnemiri, M. Ponomarenko, K. Egiazarian, *IEEE Signal Process. Lett.* **2022**, *29*, 1407.
- [37] D. P. Kingma, J. L. Ba, in *Int. Conf. Learning Representations (ICLR)*, San Diego, CA, USA **2015**.
- [38] A. Geiger, P. Lenz, R. Urtasun, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2012**.
- [39] M. Ponomarenko, N. Gapon, V. Voronin, K. Egiazarian, *Electron. Imaging* **2018**, *2018*, 382.
- [40] Udacity, <https://github.com/udacity/self-driving-car> (accessed: 01 2023).
- [41] P. Irmisch, D. Baumbach, I. Ernst, A. Börner, in *IEEE Int. Conf. Image Processing (ICIP)*, IEEE, Piscataway, NJ **2019**, pp. 1995–1999.
- [42] Allied Vision Technologies GmbH, [https://cdn.alliedvision.com/fileadmin/pdf/en/Prosilica\\_GC\\_1380H\\_DataSheet\\_en.pdf](https://cdn.alliedvision.com/fileadmin/pdf/en/Prosilica_GC_1380H_DataSheet_en.pdf) (accessed: 01 2023).
- [43] Ximea GmbH, <https://www.ximea.com/en/products/cameras-filtered-by-sensor-types/mq013rg-e2> (accessed: Januray 2023).
- [44] G. Chen, F. Zhu, P. Ann Heng, in *Int. Conf. on Computer Vision (ICCV)*, Santiago, Chile **2015**.
- [45] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, *IEEE Trans. Image Process.* **2007**, *16*, 2080.
- [46] A. Buades, B. Coll, J.-M. Morel, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, IEEE, Piscataway, NJ **2005**, pp. 60–65.
- [47] Z. Wang, J. Liu, G. Li, H. Han, in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, IEEE, Piscataway, NJ **2022**, pp. 2027–2036.
- [48] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *IEEE Trans. Image Process.* **2004**, *13*, 600.