Camera Self-Health-Maintenance by means of Sensor Artificial Intelligence

vorgelegt von
Maik Wischow, M.Sc.

an der Fakultät IV - Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktor der Ingenieurwissenschaften
-Dr.-Ing.-

genehmigte Dissertation

Promotionsausschuss:

Vorsitzender: Prof. Dr. Marc Alexa
Gutachter: Prof. Dr. Guillermo Gallego
Gutachter: Prof. Dr. Peter Eisert
Gutachter: Prof. Dr. habil. Rudolph Triebel

Tag der wissenschaftlichen Aussprache: 07.11.2023

Berlin 2024

Acknowledgements

First and foremost I would like to express my deep gratitude to Dr. Anko Börner for the opportunity to carry out my doctoral thesis at the German Aerospace Center (DLR) in his team. Anko has been a proverbial bastion of calm during my doctoral days. With his encouraging, patient and sincere nature, he constantly supported me with help and advice, no matter how absurd my questions sounded or how busy his schedule was.

The same goes for my mentor and supervisor at the TU Berlin, Prof. Dr. Guillermo Gallego. You got my very deepest appreciation and I have always considered it an honor to work under your guidance. You have given me freedom to follow my own path, yet helped me in all circumstances, encouraged me to continuously push my limits, and offered me a warm place in the Robotic Interactive Perception team. In this regard, I would also like to thank Prof. Dr. Sergio Lucia, who took over my first supervision before moving to Dortmund.

From the Real-Time Data Processing team at the DLR, I would like to especially thank Dr. Patrick Irmisch and Ines Ernst for their wholeheartedly support since the beginning of my time in the team and for always having an open ear. My thanks goes also to Dr. Henry Meißner for his advice on my topic, André Choinowski and Dennis Dahlke for repeatedly building and preparing the necessary measuring equipment, and the entire team for having a great time. Many thanks also to Florian Rosenzweig for the amazing time and for bringing me to the DLR.

Last but not least, I would like to thank my mother Petra Corswandt, my grandparents Emma and Wolfgang Corswandt, my brother Erik Wischow and his partner Peggy Ehlers for always being by my side and always wanting the best for me. Moreover, my sincere gratitude goes to Axel, Birgit, Dennis, and Jennifer Waniczek, who warmly welcomed me into their family, and have given me joy, strength, and backing throughout the years. This goes primarily to my love, who has given me a lot of hold and support in tough times. I further consider my lifelong friend Tobias Gürtler as part of my family, appreciate his constant support and all the time we have spent together.

Abstract

Autonomous machines require increasingly more robustness and reliability to meet the demands of modern tasks. These requirements specially apply to cameras onboard such machines, as they are the predominant sensors acquiring information about the environment to support decision making and actuation. Hence, the cameras must maintain their own functionality. This poses a significant challenge, primarily driven by the variety of existing cameras, the vast amount of potential application scenarios, and the limited machine resources, all while demanding real-time performance. Existing solutions are typically tailored to specific problems or detached from the downstream computer vision tasks of the machines, which, however, determine the requirements on the quality of the produced camera images.

This thesis presents a camera self-health-maintenance framework to bridge this gap. The approach combines a generalized condition monitoring and a task-oriented decision & control unit. The monitoring is based on novel learning-based blur and noise estimators that incorporate physical knowledge about the camera to increase consistency and robustness. Especially the incorporation of camera metadata enables the system to disambiguate the contributions of different noise processes within a camera. In this manner alone, the decision & control unit can initiate appropriate countermeasures, if necessary. To this end, camera parameters are readjusted based on an empirical image task analysis to optimize performance under any situation.

The framework is evaluated on synthetic and real datasets from transportation and robotic scenarios in terms of accuracy, robustness and real-time capability. Firstly, the blur and noise estimators are examined and two extensions are analyzed, which recover the estimation of combined blur/noise corruptions and reduce estimation uncertainties, respectively. Secondly, the effect of an acquired image and the camera's metadata on noise source estimation is investigated. This method is further demonstrated on the detection of mismatches between both inputs (image and camera metadata) to quantify unexpected noise as from camera defects. Lastly, the framework is implemented and verified on a real robot system.

The real demonstration on a robot shows promising results to employ the framework for arbitrary mobile machines in unknown environments. In particular, the proposed framework outperforms standard camera parameter controllers. Yet, the results also highlight current limitations that require framework extensions in future studies, such as the application to complex non-linear motion blur and scenes with high dynamic light intensity ranges.

Zusammenfassung

Autonome Maschinen benötigen zunehmend mehr Robustheit und Zuverlässigkeit, um den Anforderungen moderner Aufgaben standzuhalten. Diese Anforderungen gelten insbesondere für Kameras an Bord solcher Maschinen, da sie die vorherrschenden Sensoren sind, die Informationen über die Umgebung erfassen und somit die Entscheidungsfindung und Steuerungsprozesse unterstützen. Aus diesem Grund müssen die Kameras ihre eigene Funktionalität aufrechterhalten. Dies stellt eine große Herausforderung dar, die in erster Linie durch die Vielfalt der vorhandenen Kameras, die große Anzahl potenzieller Anwendungsszenarien und die begrenzten Maschinenressourcen bedingt ist, während gleichzeitig Echtzeit-Performance benötigt wird. Derzeitige Lösungsansätze sind in der Regel auf spezifische Problemstellungen zugeschnitten oder losgelöst von den nachgelagerten Bildverarbeitungsaufgaben der Maschinen, die jedoch die Anforderungen an die Qualität der erzeugten Kamerabilder bestimmen.

In dieser Arbeit wird ein Framework zur Selbstinstandhaltung von Kameras vorgestellt, um diese Lücke zu schließen. Das Framework kombiniert eine Zustandsüberwachung mit einer Entscheidungs- & Steuerungseinheit. Die Zustandsüberwachung basiert auf neuartigen lernbasierten Unschärfe- und Rauschschätzern, die physikalisches Wissen über die Kamera einbeziehen, um die Konsistenz und Robustheit zu erhöhen. Insbesondere Kamera-Metadaten ermöglichen es, die Beiträge verschiedener Rauschprozesse innerhalb einer Kamera voneinander abzugrenzen. Nur so kann die Steuereinheit bei Bedarf geeignete Gegenmaßnahmen einleiten. Zu diesem Zweck werden die Kameraparameter auf Grundlage einer empirischen Analyse der Bildverarbeitungsaufgaben nachjustiert, um die Performance je nach Situation zu optimieren.

Das Framework wird anhand von synthetischen und realen Datensätzen aus Transport- und Roboterszenarien in Hinblick auf Genauigkeit, Robustheit und Echtzeitfähigkeit evaluiert. Zunächst werden die Unschärfe- und Rauschschätzer analysiert und zwei Erweiterungen untersucht, die die Schätzung von kombinierten Unschärfe-/Rauschverfälschungen wiederherstellen beziehungsweise die Schätzungsunsicherheiten verringern. Zweitens werden die Auswirkungen eines aufgenommenen Bildes und der Metadaten der Kamera auf die Schätzung der Rauschquelle untersucht. Diese Methode wird außerdem bei der Erkennung von Abweichungen zwischen beiden Eingaben (Bild und Kamerametadaten) demonstriert, um unerwartetes Rauschen zu quantifizieren, das zum Beispiel auf Kameradefekte zurückzuführen ist. Abschließend wird das Konzept auf einem realen Robotersystem implementiert und verifiziert.

Die Demonstration am realen Roboter liefert vielversprechende Ergebnisse für den Einsatz des Frameworks bei beliebigen mobilen Maschinen in unbekannten Umgebungen. Insbesondere übertrifft das vorgeschlagene Framework die Standard-Kameraparametersteuerung. Die Ergebnisse weisen jedoch auch auf aktuelle Einschränkungen hin, die in zukünftigen Studien Erweiterungen des Frameworks erfordern, wie zum Beispiel die Anwendung auf komplexe nichtlineare Bewegungsunschärfe und Szenen mit hohem Dynamikumfang der Lichtverhältnisse.

Table of Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description
Physical Quantities (General)	
e^-	Electron
f	Frequency
k	Boltzmann constant
q	Electric charge
v	Velocity
C	Capacitance
N_{e^-}	Number of electrons
T	Temperature
V	Voltage
Physical Quantities (Sensor System)	
d_p	Pixel pitch of a camera sensor
f_c	Flicker noise corner frequency
γ	Noise type control variable
τ_D	Correlated double sampling dominant time constant
τ_{RTN}	Random telegraph noise characteristic time constant
t_s	Correlated double sampling sample-to-sample time
$u(x, y)$	Intensity of a noise process u at coordinates (x, y)
f_{clock}	Readout frequency
A_{ADC}	Analog-to-digital converter gain
A_{SF}	Source follower gain
A_{SN}	Sense node gain
D_{FM}	Dark current figure-of-merit at $T = 300\,\mathrm{K}$
ΔI	Induced source-follower current modulation
$E_g(T)$	Temperature (T) dependent band gap energy of a semiconductor
E_{g0}	Material depending term of a band gap energy E_g
H_{CDS}	Correlated double sampling transfer function
S_{RTN}	Random telegraph noise power spectrum
S_{SF}	Source-follower noise power spectrum
$S_{\widehat{\mathrm{DC}}}$	Dark signal S with expected dark current $\widehat{\mathrm{DC}}$

Symbol	Description
Physical Quantities (Sensor System, cont.)	
V'	By source follower amplified voltage V
$V_{[\cdot].\mathrm{Ref}}$	Reference voltage
W	White noise power spectrum
Physical Quantities (Lens System)	
(c_x, c_y)	Defocus blur center
f	Focal length
r	Defocus blur radius
s	Defocus blur normalization variable
d_B	Out-of-focus distance of a scene object
d_I	Focus distance of the camera sensor plane
d_O	Focus distance of a scene object
D_A	Aperture diameter
D_I	Projected scene object point size
H	Hyperfocal distance
L	Motion blur length on the camera sensor plane
Camera System	
$c1, c2$	Camera systems
t_{exp}	Camera exposure time
$M_{[\cdot]}$	Set of camera metadata
$M'_{[\cdot]}$	Altered set of camera metadata
Images and Image Processing	
b	Blur kernel
d	Blur kernel size
f_{Nyquist}	Nyquist frequency
f_s	Number of alternating black-white Siemens star segments
h, w	Image height and width
$h(x, y)$	Intensity of a blur kernel h at coordinates (x, y)
σ	Noise level (standard deviation of image intensities)
σ_σ	Standard deviation of a standard deviation
$\xi_{\mathrm{M/I}}$	Residual noise level
t	Image time stamp
A_n	Aggregation window size
C	Image contrast
C_0	Maximum possible image contrast modulation
$\Delta_{\sigma_{a\to b}}$	Change of a std. dev. σ between aggregation windows sizes a and b

Symbol	Description
Images and Image Processing (cont.)	
$I_{\text{Min}}/I_{\text{max}}$	Minimum and maximum image intensities
$I(x, y)$	Intensity of an image I at coordinates (x, y)
$I^{\star}(x, y)$	Intensity of an image I at coordinates (x, y) corrupted by blur
$\tilde{I}(x, y)$	Intensity of an image I at coordinates (x, y) corrupted by noise
MB_L	Motion blur length
N	Number of image frames
Object Detection	
p	Object detection confidence score
t_{FW}	Execution time of the proposed framework (FW)
B_D	Object detection
B_{GT}	Ground truth object detection
Machine Learning and Statistics	
x_{max}	Coordinate of the maximum of a statistical distribution
G, R	Prior knowledge terms
$\mathcal{L}$	Total loss function
$\mathcal{L}_{\text{NN}}$	Loss function of a neural network
$\mathcal{L}_{\text{phys}}$	Physical regularization term of a loss function
$\mathcal{N}(\mu, \sigma^2)(x)$	Gaussian distribution $\mathcal{N}$ with mean μ, variance σ^2, and input x
$\mathcal{P}(\lambda)(x)$	Poisson distribution $\mathcal{P}$ with expected value λ and input x
Notations	
λ, α, β	Positive scalars
ϵ	Small positive scalar (small error)
$f(\cdot),\ g(\cdot)$	Unspecified (mock) functions
θ, Φ	Angles
$(t)_p$	For a first-order logic formula t, the variable p is assumed to be fixed.
$[a, b]$	Interval between real numbers a and b (including border values)
$[\cdot]_{jk}$	Matrix element at coordinate (j, k)
$[\cdot]^{(1)}, \ldots, [\cdot]^{(n)}$	Consecutive frames 1, ..., n
$\{\cdot\}_{i=0}^{N}$	Set of N quantities
$\cup_{jk}\{[\cdot]_{jk}\}_{i=0}^{N}$	Union over sets of matrix elements at coordinate (j, k)
$[\cdot]^{\star}$	Optimized variable
$[\cdot]_{\text{norm}}$	Normalized quantity
$\bar{[\cdot]}$	Estimated average quantity
$\hat{[\cdot]}$	Estimated quantity
$\tilde{[\cdot]}$	Estimated median quantity
$\overset{\mathcal{F}}{\mapsto}$	Fourier transformation

Acronyms

Acronym	Description
AI	Artificial Intelligence
AMAE	Average Mean Absolute Error
AP	Average Precision
AWGN	Additive White Gaussian Noise
B+F	Blurring and Filtering (noise estimation)
BM3D	Block-Matching and 3D Filtering (denoiser)
CCD	Charge-Coupled Device
CDS	Correlated Double Sampling
CIS	CMOS Imaging Sensor
CMOS	Complementary Metal-Oxide-Semiconductor
CNN	Convolutional Neural Network
CoC	Circle of Confusion
CPU	Central Processing Unit
CTF	Contrast Transfer Function
DC	Dark Current
DCSN	Dark Current Shot Noise
DEFCARS	DEFocused CARS (dataset)
DN	Digital Number
DNN	Deep Neural Network
DoF	Depth of Field
DRNE	Deep Residual Noise Estimator
ESF	Edge Spread Function
FBI	Fast Blind Image (denoiser)
FCB	Fully Connected Branch
FLOP	FLoating point OPerations
FLOP/s	FLoating point OPerations per second
FW	FrameWork
GAN	Generative Adversarial Network
GBB	Graph-Based Blur Estimation
GPT	Generative Pre-trained Transformer

Acronym	Description
GPU	Graphics Processing Unit
GT	Ground Truth
IOPC	Input-Output Performance Curves
IOU	Intersection Over Union
IQA	Image Quality Assessment
IQR	InterQuartile Range
IR	InfraRed
LinMB	Linear Motion Blur
LSF	Line Spread Function
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAP	Maximum-A-Posteriori
MB	Motion Blur
ML	Machine Learning
MOTCARS	MOTion blurred CARS (dataset)
MTF	Modulation Transfer Function
NLF	Noise Level Function
NLM	Non-Local Means (denoiser)
NN	Neural Network
OTF	Optical Transfer Function
PCA	Principal Component Analysis (noise estimation)
PDE	Partial Differential Equation
PGE-Net	Poisson-Gaussian Estimation Net
PhTF	Phase Transfer Function
Physics-ML	Physics-informed Machine Learning
PMP	Patch-wise Minimal Pixels (blur estimation)
PN	Photon shot Noise
PrRe	Precision-Recall-curve
PSF	Point Spread Function
PSNR	Peak Signal-to-Noise Ratio
R-CNN	Region-based Convolutional Neural Network
RN	Readout Noise
Sensor AI	Sensor Artificial Intelligence
SGEMM	Single-precision GEneral Matrix Multiply
SHAP	SHapley Additive exPlanations
SLAM	Simultaneous Localization And Mapping
SLE	SLanted Edge

Acronym	Description
SN	Sense Node
SSIM	Structural Similarity Index Measure
TP	True Positive
UV	UltraViolet
VO	Visual Odometry
YOLO	You Only Look Once

CHAPTER 1

Introduction

Machines are indispensable to facilitate and automate tedious, time-consuming or dangerous tasks. Today, we are on the verge of the 5th industrial revolution, which is characterized by an increasing use of machines that evolve away from manual control towards autonomy and can act independently of time and place. This evolution is equally increasing the complexity of the machines and with it the need for reliability and robustness to ensure the safety of people and the machines themselves. These requirements run like a red thread through all system components, starting with the sensors that perceive the environment. Special attention is paid to trustworthy perception, as all subsequent actions depend on it. Cameras are nowadays the predominant sensors to perceive the environment, and are therefore the subject of this thesis. To guarantee a camera's intended functionality, autonomy also demands for self-health-maintenance, meaning the task of continuously monitoring the behavior of the system and executing automatic countermeasures in case of a detected misbehavior [Wis+23b].

For this purpose, various automatic image quality maintenance techniques have been developed and are now part of a standard camera's imaging pipeline (auto-focus, auto-exposure, auto-calibration, etc.). However, such techniques tend to be decoupled from the envisaged high-level image application and may therefore not achieve optimal whole-system performance. This applies especially where image quality may be traded off against other high-level application benefits. Moreover, every high-level application has its own criteria for what is considered an optimal image quality.

The requirements for reliability and robustness likewise apply to the camera self-health-maintenance and substantially depend on the choice of methodology. Modern artificial intelligence (AI) algorithms feature high generalizability and flexibility, which makes them the first choice for complex tasks and unknown environments. Yet they are still considered to be incomprehensible in their decision-making processes, unpredictable in their behavior, data-dependent, and vulnerable to attack. On the other hand, hand-crafted traditional methods offer a comprehensive understanding of the problem being solved but lack generalizability and flexibility.

In this thesis, a general self-health-maintenance framework is proposed that strives for optimal application performance and reliable/robust operation. To this end, novel AI-based camera condition monitoring approaches are developed that incorporate the physical knowledge of the camera sensor system (Sensor AI), and a decision & control policy that readjusts the camera configuration based on the camera's current state and its targeted high-level task. All of the framework's components are presented on a theoretical basis, evaluated in extensive experiments, and implemented in two real-world camera systems in order to provide new insights to the research area and a platform for its application on modern machines.

1.1 Research Focus

This section describes the objectives of this thesis, its scope, and derived research questions.

Objectives

The aim of this study is to design and evaluate a (*i*) practical (*ii*) self-health-maintenance framework for camera systems in autonomous machines by means of (*iii*) Sensor AI.

(*i*) The target framework is considered practical if it is generalizable to different camera systems, can be executed on mobile hardware in real-time, and operates reliably.

(*ii*) Self-health-maintenance implies the tasks of continuously monitoring the behavior of the system and executing automatic countermeasures in case of a detected misbehavior. In order to evaluate the behavior and thus distinguish a good condition from a bad one, this thesis builds upon previous work by linking the systems' condition to the quality of the data it produces. However, the assessment of image quality depends on its "intended marketplace or application" [BPA02] – hence, this thesis investigates the task of monitoring image quality with respect to an envisaged high-level application. Moreover, automatic countermeasures have to be chosen to alter image quality in a way that optimizes the performance of the target application. For the sake of finding appropriate countermeasures, the proposed framework is also intended to identify the root causes for undesirable image conditions.

(*iii*) The last focus lies on investigating combinations of data-driven and physics-based approaches that unite their best properties to robustly and reliably perform the sub-tasks of image quality assessment and determine root causes for image degradations. However, in line with Sensor AI, we adopt a "holistic approach which considers entire signal chains from the origin to a data product" [Bör+20].

Scope

Due to the vast fields of the included topics, the scope of this work is limited to

- **Field of Application:** Autonomous mobile machines in transportation and robotics scenarios on ground. The focus is on object detection as an exemplary high-level image application with great importance in these fields.
- **Camera System:** Panchromatic digital camera systems that operate in the visible electromagnetic spectrum (around 380–700 nm wavelength). For the sake of simplicity, the camera system is assumed to produce images with 8-bit radiometric resolution, i.e., intensity values in [0,255] digital numbers (DN).
- **Image Quality:** Blur and noise as two important and well-researched image quality attributes.
- **Root Causes:** Time-varying root causes of blur and noise (since any time-invariant effects are usually mitigated in an offline camera calibration), and region-wise effects (allowing to consider spatially-varying problems). This work considers only blur and noise root causes that originate in a camera system and not due to post-processing, data transmission, etc.
- **Countermeasures:** Automatic countermeasures tackling motion blur in the presence of noise, since object detectors are substantially more sensitive to blur than to noise [HD19] and motion blur is challenging to control as it also depends on moving objects in the environment.

Research Questions

Given the objectives and the scope, this thesis aims to answer one superordinate and four derived research questions:

- How can a camera system autonomously optimize the performance of a target application (object detection) that operates on the data it produces, with a focus on self-induced image blur and noise?
  - (a) How can the sources of camera noise be identified and quantified based on knowledge of the camera system's physical working principle and its produced image-/meta-data?
  - (b) How accurate and robust does the self-health-maintenance framework of the camera system perform in the context of automotive and robotic scenarios?
  - (c) How can the self-health-maintenance framework be optimized to make it run in real-time on a mobile machine with limited computational resources?
  - (d) What are the limits of the self-health-maintenance framework with respect to the machine's digital camera system and the image target application?

1.2 Contributions

The two main and two minor contributions are summarized as follows:

The first main contribution is the design and implementation of a framework to automatically maintain the intended functionality of a camera system. The framework consists of two parts: state monitoring of a camera system and automatic countermeasures in case of a detected misbehavior. The estimation part is based on objective image quality assessment (blur and noise). The countermeasure part relies on extensive sensitivity analyses of a target image application (object detection) and the implementation of a camera control routine (ISO gain and exposure time trade-off). The key advantage of the framework is its coupling to an envisaged high-level application that allows to optimize whole-system performance, while maintaining a modular design. In addition, improved machine-learning–based (ML-based) methods are proposed that incorporate physical knowledge of the camera system for blur/ noise estimation. Extensive experiments demonstrate their superior accuracy compared to traditional approaches, real-time capability, memory-efficiency, and practical recommendations for the robustness of camera monitoring applications. All framework components are further independently evaluated in experiments on a real-world ground vehicle. The source code is provided in https://github.com/MaikWischow/Camera-Condition-Monitoring.

The second main contribution is a ML–based noise source estimator that not only estimates the total amount of noise in an image, but also evaluates camera metadata to identify and quantify major noise root causes. Moreover, the noise source estimator includes a verification mechanism that quantifies noise mismatches between the metadata and the image noise, which serves for self-control and detection of unexpected events (e.g., camera damages). To the best of the authors' knowledge, this is the first estimator (traditional or learning-based) to explicitly quantify four individual noise source contributions. Extensive experiments investigate synthetic noise, real-world noise extracted from camera systems, qualitative field campaigns on a ground vehicle, the influence of each individual camera metadata, and also create unexpected noise events in images or metadata. Lastly, an improved performance on total noise estimation is demonstrated on the well-known downstream task of image denoising. The corresponding source code is provided in https://github.com/MaikWischow/Noise-Source-Estimation.

Minor contributions include (*i*) the evaluation of the estimation of combined blur and noise that interfere with each other, and (*ii*) an improved blur estimation routine in the presence of high noise.

Parts of this thesis have been published in following peer-reviewed publications:

- Wischow, Maik et al. (2022). "Calibration and Validation of a Stereo Camera System Augmented with a Long-Wave Infrared Module to Monitor Ultrasonic Welding of Thermoplastics". In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pp. 179–186.
  Chapter: 3.
- Wischow, Maik et al. (2023). "Monitoring and Adapting the Physical State of a Camera for Autonomous Vehicles". IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2023.3328811.
  Chapters: 1, 2, 3, 4, 5, 7.
- Wischow, Maik et al. (2023). "Estimating the Noise Sources of a Camera System from an Image and Metadata". Under review.
  Chapters: 2, 4, 6.

Additional non-peer-reviewed contributions:

- Wischow, Maik et al. (2020). "Camera Condition Monitoring and Readjustment by means of Noise and Blur". Talk. 1st Sensor AI Workshop. Berlin. Germany.
- Wischow, Maik et al. (2021). "How to combine Physics and Machine Learning". Talk. 2nd Sensor AI Workshop. Berlin. Germany.
- Wischow, Maik (2021). "AI-based Condition Monitoring for Cameras". Talk. Photonics Days Berlin Brandenburg 2021. Berlin. Germany.
- Wischow, Maik (2022). "Camera self-health-maintenance framework implementation for the Integrated Positioning System (C++, Python)". Software. German Aerospace Center. Berlin. Germany.
- Wischow, Maik et al. (2022). "A camera self-health-maintenance system based on Sensor Artificial Intelligence". Poster. Adlershofer Forschungsforum. Berlin. Germany.

1.3 Outline

This thesis is organized as depicted in Fig. 1.1. Chapter 1 motivates on the background that machines are becoming more and more autonomous and the camera system is an important part of a machine that needs to be maintained by itself in order to successfully accomplish a targeted task. Research objectives are stated, the scope of this thesis is defined, and contributions are summarized.

Chapter 2 reviews state-of-the-art approaches from the related fields of adaptive camera regulation (online estimation of the current vision state and automatic executions of an

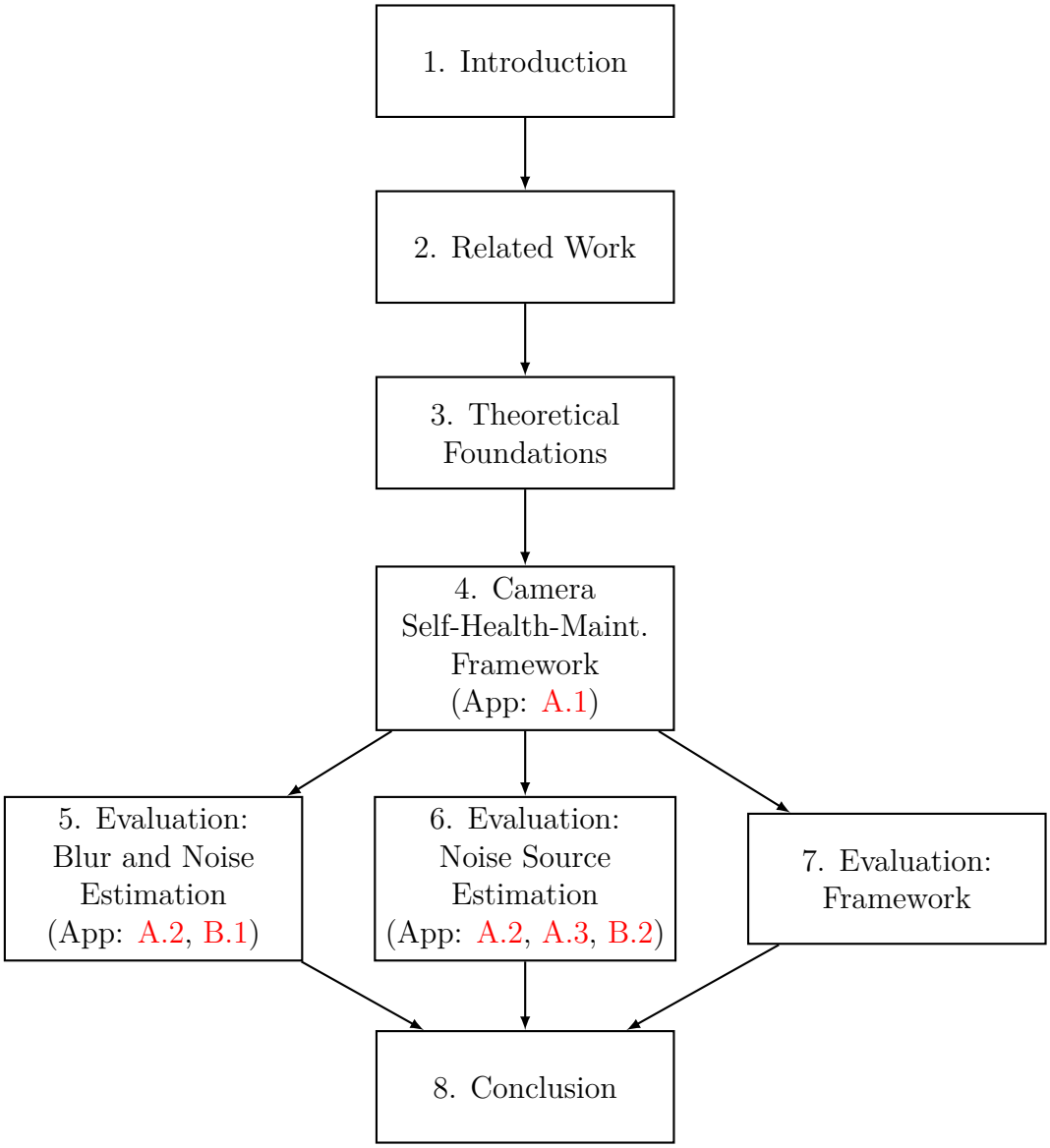


Figure 1.1: *Structure of the thesis,* according to chapters and corresponding appendices.

action to improve a target criterion, focused on tackling motion blur), image quality assessment (focused on the objective quantification of blur and noise), and physics-informed machine learning (combination of machine learning and physics, with focus on Sensor AI). On this basis, advantages and drawbacks of existing approaches are discussed and further design decisions for the proposed framework are identified.

Chapter 3 introduces the fundamental concepts related to a digital camera system, its produced data, and the assessment of image quality on which this thesis builds upon. First, the basic structure of a digital camera system is presented and theoretical models are proposed to describe its two main components: the sensor system and the lens system. Both components determine the quality of the produced data, which is the subject of the second part, where models for the concepts of blur and noise as two major image quality attributes are presented. Third, classic standard approaches to assess

blur and noise are introduced. Lastly, shortcomings of the used models, extended sensor systems, and more extensive models are discussed.

Chapter 4 describes the camera self-health-maintenance framework that is developed in this thesis. After a general overview, the condition estimation and decision & control modules are detailed, including the improvements made to existing ML-based blur and noise estimators, the working principle of traditional estimators, the introduction of noise source estimation, and the automatic camera parameter readjustment on the basis of learned input-output performance curves for blur, noise, and object detection. Alternative designs as well as future extensions are discussed for each component separately. Finally, limitations of the framework's applicability are addressed.

Chapter 5 covers the evaluation of the proposed blur and noise estimators. For this, employed datasets with synthetic and real-world corruptions are presented first. Subsequently, the blur and noise estimators are evaluated on respective isolated and simultaneously occurring corruptions. On this basis, two improvements are introduced: blur estimation in the presence of high noise and noise estimation with reduced uncertainty. Finally, limitations and further potential improvements are discussed.

The next Chapter 6 is dedicated to the evaluation of the proposed noise source estimation. First, the synthetic and real-world datasets used and the image noise applied are specified. The following sections contain the evaluations on quantitative and qualitative experiments. Furthermore, it is demonstrated how noise source estimation is applied in real field campaigns to detect mismatches between image and metadata noise, and to denoise real-world images. The chapter is wrapped up with a discussion on shortcomings and potential extensions.

Chapter 7 demonstrates the combination of online blur/ noise estimators and offline empirical input-output performance curves for practical application to control image quality and hence optimize the system's performance. Therefore, exemplary performance curves that relate object detection performance to different blur and noise levels are first determined. Thereafter, synthetic and real-world scenarios are proposed on which the application of the framework is then demonstrated. Finally, the framework's required computational costs on stationary and mobile hardware are examined, and further details and methological drawbacks are discussed.

The final Chapter 8 summarizes the answers to the research questions, gained insights, and the proposed starting points for future studies.

CHAPTER 2

Related Work

The core of this thesis is closely related to *active vision* [AWB88], *adaptive camera regulation* [MFR96], and *camera attribute control* [Han+23] in that there are two connected tasks: online *estimation* of the current vision state and automatic execution of an *action* to improve some target criterion. In the estimation task, we estimate major properties of the camera system state by assessing the quality of the image data *it produces* in terms of blur and noise. Subsequently, we define actions that can be carried out to control the camera, therefore influence image properties (we demonstrate this for motion blur and noise) and hence optimize the system's performance for a target application (object detection in this work). This pipeline shall be underpinned by Sensor AI techniques to take advantage of both data-based and physics-based concepts.

We first review general adaptive camera regulation approaches and ones that address motion blur in particular (Sec. 2.1). Following this, we survey approaches for automatic blur and noise assessment (Sec. 2.2). Next, we outline approaches from the field of physics-informed ML with a focus on its subtopic Sensor AI (Sec. 2.3). Finally, we discuss and summarize this chapter (Secs. 2.4 and 2.5).

2.1 Adaptive Camera Regulation

Section 2.1.1 provides an overview to the field of adaptive camera regulation and subsequently Sec. 2.1.2 focuses on studies that target motion blur. Table 2.1 summarizes the main works that are closest to this thesis.

2.1.1 Overview

Several works study automatic camera parameter readjustments in order to improve the performance of various high-level image tasks of mobile machines, such as object detection [Shi+19; Mud+19; SM19; LMM21; Muk+21; OMH21; SWM21], object tracking [SMM18; Mud+19], action detection [Mud+19; LMM21], visual odometry (VO) [Shi+18; Tom+21; Han+23], simultaneous localization and mapping (SLAM) [Kim+17; Shi+18; Tom+21], feature detection and mapping [Kim+17; Shi+18; Shi+19; Tom+21],

Study	Approach	Image Task	Image Features	Camera Parameters	Motion Blur
Saha et al. [SMM18]	DNN	Obj. Det.	Learned	Spat. Res.	No
Saha et al. [SM19]	Reinf. Learn.	Obj. Det.	Learned	Spat. Res.	No
Mudassar et al. [Mud+19]	DNN	Obj. Det., Act. Det.	Learned	Spat. Res., Temp. Res.	No
Lee et al. [LMM21]	DNN	Img. Class., Obj. Det., Act. Det.	Learned	Spat. Res., Voltage	No
Tomasi et al. [Tom+21]	DNN	(SLAM/VO)	Feature Points	Exp. Time, Gain	No
Onzon et al. [OMH21]	DNN	Obj. Det.	Learned	Exp. Time	No
Torres and Menéndez [TM15]	Traditional	Surveillance	Intensity	Exp. Time, Gain	No
Kim et al. [Kim+17]	Traditional	Self. Localiz.	Intensity	Exp. Time	No
Shim et al. [Shi+18]	Traditional	Obt. Det., VO	Gradients	Exp. Time, Gain	No
Shin et al. [Shi+19]	Traditional	Obt. Det., Feat. Match., Pose Est.	Greadients, Entropy, Noise	Exp. Time, Gain	No
Westerhoff et al. [WMK15]	Traditional	-	Intensity, Entropy	Generic	No
Oktay et al. [OCT18]	Traditional	-	-	-	Yes
Wang et al. [WLR23]	Traditional	Obj. Det.	-	(Robot Control)	Yes
Kim et al. [KCK18]	Traditional	SLAM	Gradients, Illumination, Noise	Exp. Time, Gain	Yes
Han et al. [Han+23]	Traditional	-	Gradients, Entropy, Opt. Flow	Exp. Time, Gain	Yes

Table 2.1: *Summary of main related adaptive camera regulation studies.* The table characterizes each study in terms of its type of approach (traditional or learning-based), image analysis task, image features used, camera parameters aimed for optimization, and specific consideration of motion blur.

surveillance [TM15], and the task of reducing energy consumption of a machine [SMM18; Muk+21; SWM21].

To this end, related approaches optimize against traditional hand-crafted and automatically learned image features as metrics. Traditional image features include feature points [Kim+17; SWM21], gradients [Shi+18; Shi+19; Han+23], intensity entropy [WMK15; Shi+19; Han+23], intensity histogram [TM15], noise [Shi+19], and over- or underexposure [WMK15]. In contrast, the studies [SMM18; Mud+19; SM19; LMM21; Muk+21; OMH21; Tom+21] rely on learned features.

Readjusted camera parameters contain exposure time [TM15; WMK15; Kim+17; Shi+18; Shi+19; OMH21; Tom+21; Han+23], camera gain [TM15; Shi+19; Tom+21; Han+23], pixel voltage [LMM21], adaptive regions of interest [SMM18], spatial resolution [Mud+19; LMM21; SWM21], spectral modality [Mud+19; SM19], and temporal resolution [Mud+19].

Let us summarize the key aspects of the studies that are closest to this thesis. Saha et al. propose a supervised deep neural network (DNN) [SMM18] and a reinforcement learning approach [SM19] to control the spatial modality of a visual and an infrared (IR) sensor on the basis of online feedback of an object detector in order to reduce redundant task-critical information of both sensors. As a result, they demonstrate increased object tracking accuracy and reduced energy consumption in an edge device. Analogously, Mudassar et al. [Mud+19] achieve similar improvements with a DNN for spatio-temporal resolution control of a visual/ IR sensor system using the output of object detection and action detection tasks. From the same working group, Lee et al. [LMM21] use a DNN to determine task failures due to perturbated sensor data, and control the spatial resolution and the voltage of their visual camera accordingly. Similar to the previous works, object detection performance, action detection performance, and the sensor's energy consumption could be improved. Torres and Menéndez [TM15] control exposure time and gain of a surveillance camera to optimize its dynamic range in the entire image. Kim et al. [Kim+17] quantify the effect of changing illumination in the context of self-navigation of a flying robot, and accordingly select appropriate environment maps and exposure times at runtime. An top of illumination-robustness, Shim et al. [Shi+18] further investigate brightness consistency between multiple cameras of a robot for stereo matching and demonstrates the effect on object detection and VO, among others. Shin et al. [Shi+19] also focus on illumination control based on exposure time and gain readjustments, which improves performances of object detection, feature matching, and pose estimation in their experiments. Tomasi et al. [Tom+21] research automatic exposure time and gain control by means of a DNN that maximizes the number of image features for VO or SLAM tasks, and show the effectiveness in a transportation scenario. Similarly, Onzon et al. [OMH21] employ a DNN to learn exposure times on the basis of object detection performance. Westerhoff et al. [WMK15] approach a generic automatic camera parameter control from an abstract point of view and propose a process that consists of an online stage (image acquisition with different parameter sets) and an offline stage (parameter optimization with regard to some image quality criterion for a particular image application). They examine their pipeline on the example of image entropy maximization in a transportation scene and compare against images acquired with default camera parameters from the manufacturer.

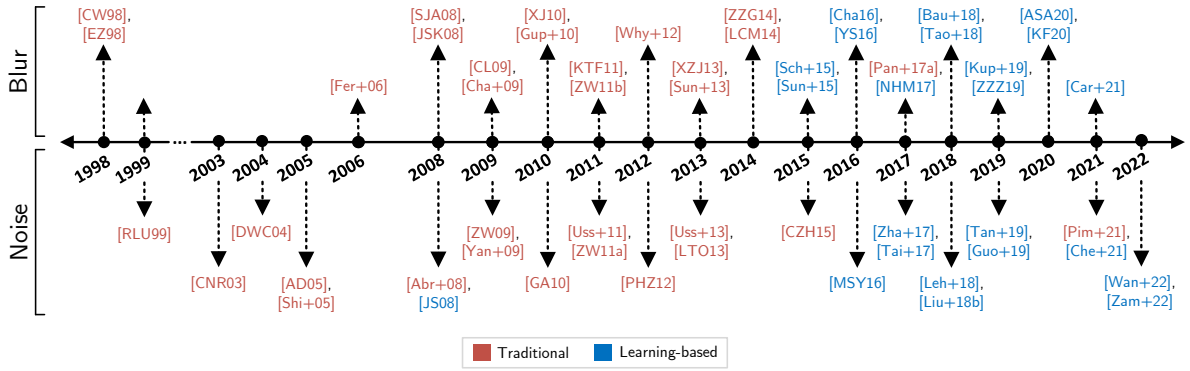


Figure 2.1: *Overview of blur and noise estimators in the literature* (limited to the two most important estimators per corruption type and year to keep it clear).

2.1.2 Motion Blur

Motion blur can be directly approached at a hardware level by involving, e.g., an accelerometer [CIS18], an inertial measurement unit [Jos+10], a self-designed sensor [HA01], lens stabilizing elements [MMV11; NS], a programmable aperture [SNT14], by shutter manipulation [RAT06], by means of multiple cameras with different configurations [BN03; LYC08; Tai+08], or by event cameras [FKS20].

Software-based motion blur regulation approaches are sparse. Oktay et al. [OCT18] and Wang et al. [WLR23] propose motion control routines for a helicopter and a quadcopter, respectively, in order to mitigate motion blur during camera exposure. Kim et al. [KCK18] demonstrate the advantage of simultaneous exposure time and gain control over a separated control to prevent motion blur and over-exposure in low-light SLAM scenarios. Specifically, isolated exposure control led to either under-exposed scenes or motion blur. Han et al. [Han+23] readjust exposure time and gain on the basis of image gradients and entropy as image quality proxies. Moreover, they incorporate the speed of the camera system (using optical flow estimations) to determine associated maximum possible exposure times that does not cause motion blur. The routine was validated in outdoor scenes, such as a garage, with a moving camera mounted on a rail to ensure a constant speed. It is worth noting, in conclusion, that "*Unfortunately, motion blur is not well considered by previous work*" [Han+23].

2.2 Image Quality Assessment

This section outlines *automatic* image quality assessment methods for *blur* and *noise* estimation (Secs. 2.2.1 and 2.2.2). Figure 2.1 provides an overview to the main studies.

2.2.1 Blur Estimation

We focus on methods able to work on a *single* frame, and are thus not restricted to the availability of *multiple* frames. They may be further classified into *blind* and *non-blind*, depending on whether the blur kernel is known. In practice the kernel is unknown. Existing blind single-frame estimation frameworks can be categorized into *traditional* or *learning-based* ones.

Traditional Approaches

The estimation of a blur kernel from only a blurred image is considered an inverse problem, which typically yields non-unique and unstable solutions [Tar05, p. xi]. Several works approach this problem by introducing prior knowledge for the underlying blur kernel or the clean image to constrain the solution space. Hence, the problem is generally addressed together with the task of image deblurring.

Prior knowledge can include assumptions about image intensities [Pan+17a; Pan+17b; Yan+17; Wen+20], gradients [SJA08; CL09; Cho+11; Sun+13; Ren+16; Bai+18; Che+19], smoothness [CW98; JSK08; SJA08; KTF11; Lev+11; XZJ13], spectral properties [LCM14; Pan+19], the image formation model [Gup+10], or be learned in a data-based manner [ZW11b; Zuo+16; Li+18; Xu+17; Ren+20; Lia+21a]. Milestones are, for instance, the studies of Pan et al. [Pan+17a; Pan+17a] (dark channel prior that assumes dark intensities in clean images but not in blurred ones), Shan et al. [SJA08] (reweighting method to avoid delta motion blur kernel solutions), Cho and Lee [CL09] (fast deconvolution using computationally efficient image filters in a multi-scale approach), Chan and Wong [CW98] (total variation prior to recover image edges), Krishnan et al. [KTF11] (fast and robust scale-invariant prior), Xu et al. [XZJ13] (L_0 prior that favors image saliency and enables fast convergence), and Zoran and Yair [ZW11b] (relies on patch models for whole-image restoration and a learned Gaussian Mixture prior).

Another categorization axis is the underlying deblurring framework. All the aforementioned works approach the problem by iteratively estimating the blur kernel and the clean image in a maximum-a-posteriori (MAP) framework. Others apply the variational Bayesian inference framework [Fer+06; TLG09; Cha+09], the split Bregman method [Cai+11; ZZG14] or other approaches [EZ98; Yit+98; XJ10; HXZ11; Why+12] to solve the optimization problem (e.g., maximum likelihood estimation with marginalization [HXZ11] or MAP combined with Richardson-Lucy deblurring [Why+12; Ric72; Luc74]).

Note that most works focus on motion-like blur kernels [SJA08; CL09; Ren+16; Sun+13; Bai+18; Lev+11; XZJ13; Pan+17a; Pan+17b; Yan+17; Wen+20; Zuo+16; Li+18; Xu+17; Ren+20; LCM14; Pan+19; TLG09; Cai+11].

Learning-Based Approaches

Learning-based methods estimate blur kernels *explicitly*, or *implicitly* within an end-to-end deblurring pipeline.

Explicit estimators base on convolutional neural networks (CNNs) [Sch+15; Cha16; Bau+18], deep auto-encoders [ASA20; KF20; Car+21], or general regression NNs [YS16; Spe+91]. Chakrabarti uses a CNN to learn Fourier representations of blur kernels [Cha16]. Schuler et al. propose a CNN with a joint kernel and clean image estimation [Sch+15]. Bauer et al. train a CNN to estimate anisotropic modulation transfer functions of a blurred image [Bau+18]. Asim et al. use generative network models with an auto-encoder architecture [ASA20]. The approach of Kaufman and Raanan relies on a U-Net [RFB15] augmented with custom dense layer operations in all convolution layers [KF20]. Carbajal et al. use an encoder and two decoders [Car+21]; similar to [Sun+15], their composite neural network (NN) estimates a set of basis kernels, but also pixel-wise coefficients to weight the influences of the blur kernels. Yan and Shao combine a DNN (to classify the blur type of an image) with a general regression NN (to subsequently regress the blur parameters) [YS16].

Implicit end-to-end pipelines leverage a wider range of models, such as conventional CNNs [Hra+15; Sun+15; APS19], multi-scale CNNs [NHM17; NCF17], deep auto-encoder-like architectures [Gon+17; NKR17; Tao+18; ZZZ19; Jia+20; PR20; SPR20], generative adversarial networks (GANs) [Kup+18; Kup+19; LCC19; Zha+20], a recurrent NN [Zha+18], and multi-scale long short-term memory (LSTM) NNs [Tao+18; Gao+19]. The most influencing works include the ones of Nah et al. [NHM17] (coarse-to-fine approach with no underlying blur kernel model assumed), Tao et al. [Tao+18] (multi-scale NN combining auto-encoder structure with residual (LSTM) blocks), and Kupyn et al. [Kup+18; Kup+19] (GANs for fast and accurate deblurring).

Most learning-based approaches are dedicated to motion blur as well [Sun+15; Cha16; ASA20; KF20; Car+21; APS19; NHM17; NCF17; Gon+17; NKR17; Tao+18; Jia+20; PR20; SPR20; Kup+18; Kup+19; LCC19; Zha+18].

2.2.2 Noise Estimation

We first survey noise level estimators (Sec. 2.2.2.1) and subsequently address noise models used in related studies (Sec. 2.2.2.2).

2.2.2.1 Noise Estimators

Analogous to blur estimation, we consider methods that assume unknown noise levels and no prior knowledge (i.e., *blind* estimation) using *single* images. These may be further divided into *traditional* and *learning-based* approaches.

Traditional Approaches

Traditional approaches estimate noise in the *spacial domain* or by means of *domain transformations*.

Spacial-based methods rely on assumptions about different image features. A common assumption is the existence of homogeneous image parts from which noise levels can be directly estimated (e.g., using robustified statistics [AD05; Shi+05; Abr+08; GA10] or image histograms [RLU99]). Such estimators can be supported by high-pass filters, since noise is high-frequency image content [CNR03; Shi+05]. Moreover, Kamble et al. [KPB19], and Amer and Dubois [AD05] propose to exclude image patches with edges. Another improvement is introduced by Uss et al. who explicitly distinguish undesired texture content from noise [Uss+11; Uss+13]. Ghazal and Amer further investigate a particle filter to reduce the search time for homogeneous areas [GA10]. Although homogeneous areas allow fast and computational efficient noise estimations, they are not suitable for high-textured images and therefore lack flexibility. Furthermore, inhomogeneous image content can be identified as noise, which is why so much effort is put into filtering it.

Transform-based approaches represent an image in a different space and assume a noise-only subspace. Widespread transformation techniques rely on principal component analysis [PHZ12; LTO13; CZH15; Khm+18], discrete cosine transformation [ZW09; ZW11a; MRR19] or discrete wavelet transformation [DWC04; Yan+09; Pim+21]. All have their own benefits with over-/ underestimation in low/ high noise and textured areas.

Learning-Based Approaches

Learning-based methods either determine the noise level *explicitly*, or *implicitly* as part of an end-to-end denoising pipeline.

Explicit representatives are the studies of Zhang et al. [Zha+17] (DNN learns a residual image that corresponds to pixel-wise noise estimations), Tan et al. [Tan+19] (similar DNN trained for signal-dependent noise), Guo et al. (plain CNN with a customized loss that includes under-estimation penalization and a total variation regularizer [Guo+19; ROF92]; and a CNN with dilated convolutions and pyramid feature fusion estimates noise level map [Guo+20]), and Byun et al. [BCM21] (U-Net estimates pixel-wise Poisson-Gaussian noise model parameters).

Implicit approaches either focus only on the denoising task [JS08; YS17; Leh+18; LGP20] or on general image restoration [MSY16; Tai+17; Liu+18b; Che+21; Lia+21b; Zam+21; Wan+22b; Zam+22]. Let us first consider the dedicated denoisers. Jain and Seung provide a basis and demonstrate image denoising with a plain DNN [JS08]. Yang and Sun unroll the well-known block-matching and 3D filtering (BM3D) denoising

algorithm [Dab+07] into a CNN and explicitly transfer BM3D operations into CNN layers. Lehtinen et al. demonstrate a simple DNN denoiser without the need for clean images during training [Leh+18]. Lyu et al. propose a generative adversarial network for the denoising task [LGP20]. General image restoration networks base on CNNs with encoder-decoder architectures [MSY16; Zam+21], long short-term memory-blocks [Tai+17], wavelets as input [Liu+18b], and different transformer-block approaches [Che+21; Lia+21b; Wan+22b; Zam+22].

2.2.2.2 Noise Models

Driven by Space camera systems, extensive noise models on a subatomic level have been developed in recent decades [HK94; Jan01; KW14]. However, applications on Earth tend to employ simpler models, as follows.

The majority of research presumes an additive white Gaussian noise (AWGN) source [Shi+05; Uss+11; CNR03; DWC04; PHZ12; CZH15; Zha+17; JS08]. Given the influence of light on camera noise [Bla+97], signal-dependent noise models have been developed considering (*i*) photon shot noise and (*ii*) noise due to camera electronics (e.g., the Poissonian-Gaussian noise model) [Foi+08; Tan+19; BCM21]. A special case is the noise level function (NLF) that characterizes the dependence of noise levels on image intensity [Liu+07; SDA15; Yan+15]. To account for non-linear camera processes that affect noise statistics [ML07], some works employ the camera response function for NLF estimation; they describe a camera's physical processing as a black-box in a function [Yao16; Yao+21].

A few noise studies break down the noise caused by camera electronics and therefore consider more than two noise sources [Wan+19; Wei+20; Zha+21; OMH21], but generally "[...] *noise sources caused by digital camera electronics are still largely overlooked, despite their significant effect on raw measurement*" [Wei+20]. The works [Wei+20; OMH21] 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 [Wan+19] and [Zha+21] 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 [Che+18; ABB19; Cha+20] for synthesis, but [Zha+21] shows that "*The DNN-based [noise generators] still cannot outperform physics-based statistical methods*".

2.3 Physics-Informed Machine Learning

The research field of blending data-based methods with traditional scientific models is referred to as *theory-guided data science* [Kar+17b] and, since a vast amount of data-based models rely on machine learning nowadays, also as *scientific machine learning*

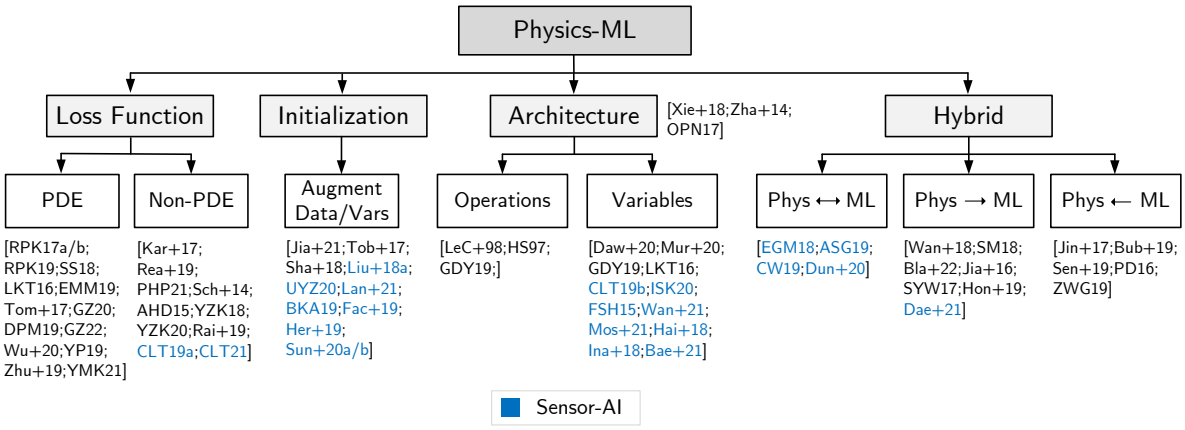


Figure 2.2: *Overview of physics-informed machine learning approaches in the literature.* Studies are classified based on their largest overlap with a category of [Wil+22].

[Cuo+22] or *physics-informed machine learning* (Physics-ML) [Kar+21]. This section is dedicated to Physics-ML only. A related field that deals specifically with the physics of sensor systems is called *sensor artificial intelligence* (Sensor AI) [Bör+20].

We first provide an overview to Physics-ML (Sec. 2.3.1) and subsequently address Sensor AI approaches with focus on optical sensor systems (Sec.2.3.2). Figure 2.2 provides an overview of Physics-ML and Sensor AI literature following to the classification of Willard et al. [Wil+22].

2.3.1 Overview

The rapidly growing field of Physics-ML is subject of multiple recent surveys [Kar+17b; Alb+19; Arr+19; RS20; Yua+20; Kas+21; Kar+21; Cuo+22; Wil+22]. The authors categorize studies according to various aspects, such as the way physics and ML are combined, the underlying scientific task, or the application domain. Our main structure follows the first approach with the same categories as Willard et al. [Wil+22]: *physics-guided loss function*, *physics-guided initialization*, *physics-guided design and architecture*, and *hybrid modeling*.

Physics-Guided Loss Function

In this strategy prior knowledge is added to the ML training process in the form of a physics loss term $\mathcal{L}_{\text{phys}}$ to penalize predictions that do not satisfy desired physical properties:

$$\mathcal{L} = \mathcal{L}_{\text{NN}}(\text{target}, \text{pred}) + \lambda\, \mathcal{L}_{\text{phys}}(\text{pred}), \tag{2.1}$$

with $\mathcal{L}$ being the new physics-reinforced training loss, the initial NN training loss $\mathcal{L}_{\text{NN}}$, and a positive weight factor λ. In the context of Bayes' theorem, the data-fidelity term $\mathcal{L}_{\text{NN}}$ can be considered as the likelihood and $\mathcal{L}_{\text{phys}}(\text{pred})$ as the prior. In the context of Bayesian [Dev11, p. 79] Note that the dependency of $\mathcal{L}_{\text{phys}}$ to predictions only does not limit this strategy to supervised scenarios. This flexibility enables the usage of general

domain-specific relationships. On the other hand, $\mathcal{L}_{\text{phys}}$ does not enforce consistency to the physics and is hence considered a soft constraint.

Pioneering work for modern approaches was done by Raissi et al. who established a framework to solve and discover general nonlinear partial differential equations (PDEs) with DNNs [RPK17b; RPK17a; RPK19]. Specifically, they propose a general physical loss term template applicable to several PDE problems. Sirignano and Spiliopoulos further enable solving high-dimensional PDEs with mesh-free data sampling [SS18]. However, the idea to solve PDEs with (shallow) NNs is not new and can be traced back to the studies [LLF98; LK90; LLP00]. Recent studies also address specific PDEs such as turbulence modeling of fluid dynamics related to the Navier-Stokes equations [LKT16; Tom+17] or investigate particular DNN models like auto-regressive networks [GZ20], auto-encoders [EMM19], echo state networks [DPM19], and transformer models [GZ22].

Physical loss constraints are demonstrated for non-PDE problems, such as lake temperature modeling [Kar+17a; Rea+19] (which employ water depth-temperature relationships) and robot dynamics modeling [PHP21] (which encodes concepts like object rigidity and inelastic collision).

From a conceptual point of view, physical loss is further used in reinforcement learning [Sch+14; AHD15], generative modeling [Wu+20; YZK18; YZK20], inverse modeling [Rai+19; Kah+20], and uncertainty quantification [YP19; Zhu+19; YMK21].

Physics-Guided Initialization

This strategy is based on the motivation that NN weights are often randomly initialized and a poor set of initial weights can lead the training procedure to local minima. A customized initialization can not only improve the result, but also accelerates the training by reducing the number of training samples required to learn a desired (physical) relationship hidden in the data. One way to incorporate such prior knowledge into the NN is to pre-train the model on a related task and fine-tune it for the desired task. Analogously, the model can be pre-trained on cheap simulated data or unsupervised to learn a domain-specific data distribution, and fine-tuned on expensive real-world data. Other ways to integrate physical knowledge are to create the NN's training data using a physics-based model or by a domain expert.

A recent prominent example is the generative pre-trained transformer (GPT) and its application for language modeling (ChatGPT) [Rad+18; Rad+19; Bro+20]. The underlying idea is to train a generative and task-agnostic NN on general text data and to fine-tune the model on a discriminative task (not limited to physical knowledge). Further well-known approaches that do not specifically involve physical knowledge, but are worth to mention in this context, are various image task NNs that base on large-scale

image data pre-training [Taj+16] (e.g., Faster R-CNN [Ren+15], YOLO [Red+16], and ResNeXt [Xie+17]).

Similar applications can be found in the fields of lake temperature modeling [Jia+21] (hidden variables in a graph NN are pre-trained), in object localization for robot grasping [Tob+17] (investigates the performance for simulated (pre-)training data only), and the pre-training of autonomous vehicles on simulated environments [Sha+18].

Physics-Guided Design and Architecture

ML algorithms still have black-box character, that is, their learned input-output relations are not interpretable by humans (or only with great effort). Physically meaningful ML learning outcomes can be ensured by adapting ML model components to the problem domain, which is like setting a hard constraint.

One way to implement this is to use physically plausible operations within the NNs. Widely used approaches in this class include CNNs [LeC+98] (which process images using the convolution operation motivated by signal processing of the human eye) and LSTM-NNs [HS97] (which enable temporal data sequences by introducing memory cells).

Other options are to: fix single NN weights or feature maps to predetermined physical variables (e.g., to water density in lake temperature modeling [Daw+20], or to pressure fields and velocity fields for fluid dynamics simulation [Mur+20]), explicitly encode physical invariances (e.g., from Hamiltonian mechanics [GDY19] or rotational invariance for turbulence modeling [LKT16]), integrate other domain-specific constraints (such as temporal coherence in fluid flow simulations [Xie+18]), and to include correlated auxiliary tasks in the learning procedure (e.g., to boost landmark detection of face images by learning additional face attributes [Zha+14] or to ensure particle properties in particle physics [OPN17]).

Hybrid Modeling

Hybrid approaches contain combinations of physics-based and ML models that are both executed simultaneously or in sequence.

The most common method is residual learning, where physics-based models provide an initial (erroneous) value and ML learns to estimate errors in form of a bias to adjust the result. This approach can be easily applied to arbitrary physical problems, but it has the disadvantage of dealing only with symptoms and therefore not targeting physically consistency. Exemplary studies apply this strategy to extreme event prediction [Wan+18], fluid dynamics modeling [SM18], and heat transfer analysis [Bla+22].

In addition to ML refinement, the results of the physical models can also serve as input to subsequent ML calculations, such as for bearing fault diagnosis [Jia+16; SYW17] (signals are compressed first and then analyzed in an auto-encoder) and remaining useful

life prediction [Hon+19] (time-frequency features are extracted from data and inputted to an ensemble of recurrent NNs).

ML can further augment physical models for ill-posed inverse problems whose solution requires a large computational effort, for instance, in sparse-view reconstruction of computer tomography images [Jin+17; Bub+19] or magnetic resonance imaging [Sen+19] (U-Nets reconstruct full-view images).

Going one step further, ML can also replace missing or erroneous terms in physical models as demonstrated in turbulent flow modeling [PD16] (radial basis functions approximate covariances) and in real-time power system monitoring [ZWG19] (recurrent NN improves computational expensive power system state estimation).

2.3.2 Sensor Artificial Intelligence

The field of sensor artificial intelligence (Sensor AI) encompasses techniques that leverage physical principles of sensor systems to improve sensor-related AI tasks, and that advance sensor systems with AI [Bör+20]. Sensor AI is similar to Physics-ML, but is tailored to sensor systems and not limited to ML (although most approaches are based on ML). In the following, we focus on optical sensor system and consider three Sensor AI categories: *AI to improve sensor systems*, *sensor systems to improve AI* and *mutual improvement of AI and sensor systems*.

AI to improve Sensor Systems

Recent studies demonstrate the ability of AI to improve the design of optical and biological sensor systems with and without specific target tasks in mind. The interested reader is referred to the surveys [Bal+21; Wet+20; Men+22] for further reading.

In the case of optical sensor systems, several components from lenses [CLT19a; CLT19b] and lens systems [CLT21] (including Cooke-Triplets or double Gauss lenses) via aperture masks for compressive sensing [San+16; ISK20] through to nano-antenna for optical communication and sensing [FSH15] have been automatically designed using AI. A special class of optical sensors for which AI-improvements are also investigated are nano-photonic-sensors [KAP15], where both Liu et al. and Unni et al. demonstrate optimization of sensor material layer thicknesses using a DNN [Liu+18a] and a mixture density network [UYZ20], respectively.

In the field of bio sensor design optimization, Joung et al. increase the cost-effectiveness of bioassays by determining an optimal subset of biotargets [Jou+19] and Valeri et al. automatically design toehold switches (programmable nucleic acid sensors) for precise diagnostics [Val+20].

Besides optical and bio sensors, Lan et al. develop dynamic meta-surface antennas for the radio frequency spectrum [Lan+21].

Sensor Systems to improve AI

Optical sensor systems have matured in the last decades and provide extensive auxiliary data that several high-level (AI) tasks can benefit from. Deardorff et al. make use of date-time, sensor pose, and temperature data from unmanned aerial systems in order to provide contextual information for hazard detection using imagery [Dea+21]. Specifically, they employ an ensemble of NNs to process image data and use the metadata to decide on an ensemble aggregation function. Extrinsic camera parameters along with intrinsics are also utilized by Wang et al. in generating training data to train an LSTM-based DNN to distinguish independent moving objects from an observed optical flow using a moving camera [Wan+21]. Bertoni et al. employ intrinsic parameters like focal length and sensor pixel size to normalize the depth of image keypoints to 3D-localize humans in single images [BKA19]. Another example to use camera intrinsics (focal length, sensor size, principal point, and pixel size) for a custom image normalization is demonstrated by Facil et al. in the context of single-view depth estimation [Fac+19]. Finally, Moseley et al. use sensor parameters as input to a CNN to estimate dark current shot noise for denoising images from low-light lunar environments [Mos+21].

Mutual Improvement of AI and Sensor Systems

Most hybrid approaches focus on improving the shape of aperture masks or diffractive optical elements in end-to-end learning frameworks that couple respective (virtual or physical) sensor components to a task-specific NN. Bacca et al. [BGA21] and Arguello et al. [Arg+23] provide an overview of different mutually improved systems.

Aperture (phase) masks has been automatically designed to extend the depth of field of conventional cameras [EGM18; ASG19], to improve single-image depth estimation [Hai+18] or color differentiation between species in microscopy [Her+19], to enable a light field acquisition from few images [Ina+18], and to compute super-resolved images from raw measurements [Sun+20a].

The study of diffractive optical element design is motivated by tasks like single-shot monocular hyperspectral depth imaging [Bae+21] and single-shot high dynamic range imaging [Sun+20b]. Free-form lens designs have been examined to improve single-shot depth estimation using coded defocus blur as an additional depth cue [CW19] and image restoration with diffractive achromatic lenses [Dun+20].

2.4 Discussion

Here we discuss advantages and disadvantages of addressed related works and derive design decisions for our self-health-maintenance framework.

Adaptive Camera Regulation

Most related studies optimize against image features, such as the number of gradients or feature points (e.g., corners), both of which serve as proxies for image quality attributes. Alternatively, one can directly optimize *image quality attributes* or *task performance* in a tailored end-to-end approach. The first option would increase interpretability (e.g., the concept of image sharpness is more intuitive compared to the number of gradients) and is task-independent, hence more general. End-to-end optimization relies on learned features that the data-driven approach considers best for task optimization and thus can lead to better task accuracy. However, learned features would be tailored to the trained task(s) only and less interpretable. Our framework aims at general and interpretable approaches, which benefits image quality over task optimization.

In terms of general adaptive camera regulation, this thesis comes closest to the work of [WMK15] in that an image database with metadata is built and analyzed for camera parameters that lead to optimal target application performance. The authors use a multi-objective optimization procedure in which the individual objectives are solved independently in a priority order (e.g., first maximize entropy and subsequently minimize amount of saturated pixels). However, this approach does not generalize because image features (more precisely, image quality attributes) can influence each other and hence should be investigated in combination (e.g., blur and noise [TL12]). Furthermore, Westerhoff et al. perform the parameter optimization offline and therefore do not consider dynamic scenes and changing camera states at runtime.

With focus on the high-level task of object detection, our work overlaps most with [OMH21]. The authors learn optimal exposure times in an end-to-end pipeline and use object detection performance from a traffic scenario as feedback. Although they incorporate an extensive image noise model, they do not account for motion blur, which is common in scenes with moving object and can be more severe than noise in object recognition tasks [DK16; HD19]. As mentioned above, such a tailored end-to-end approach is also less interpretable and flexible.

Regarding adaptive camera regulation that targets motion blur in particular, we note most similarities with [Han+23]. Han et al. readjust exposure time and gain in combination with respect to the optimization of image feature metrics at runtime. However, they neither account for the performance of the intended high-level task within the optimization, nor consider tasks that do not benefit from minimal motion blur (cf. Sec. 3.2). Both limit the generality of their framework.

Image Formation Pipeline

The majority of researchers assume simplified additive zero-mean Gaussian noise [Shi+05; Uss+11; CNR03; DWC04; PHZ12; CZH15; Zha+17; JS08], while fewer studies rely on extensive noise models close to camera physics [Wan+19; Wei+20; OMH21; Zha+21]

(which is experimentally supported as being more realistic [Xu+18; AB18]). Moreover, most works on mobile machines do not consider simultaneous blur and noise, although motion blur is to be expected especially in moving scenes; and if they do, both effects are studied in isolation.

Also note that all presented studies calibrate their noise-related parameters (temperature, exposure time, ISO gain, ...) offline and only implicitly account for changing camera parameters during training data generation, but they do not consider camera parameters at inference time. However, when it comes to noise source identification, noise parameters must be known at inference time because the problem is ambiguous when only image data is available[1].

We consider two aspects to account for these limitations. First, we use a more extensive and realistic image formation pipeline by including motion and defocus blur as well as simultaneously occurring blur and noise corruptions that influence each other. Second, we consider changing noise parameters at inference time.

Blur and Noise Estimation

We chose *explicit* and *learning-based* blur and noise estimators for five reasons related to a condition monitoring scenario. (*i*) A task-independent and interpretable framework requires explicit estimations. (*ii*) Traditional blur estimators perform kernel estimation and image deblurring in combination. The computing time needed for this joint task is not reasonable for a real time application. Moreover, in general, learning-based blur estimation methods achieve the best performance [Rim+20]. (*iii*) Traditional noise estimators rely on insufficient noise models that have been shown to be less realistic. Although the real-image denoising performance benchmark of Plotz and Roth [PR17] indicate that traditional methods perform slightly better than learning-based methods, this analysis does not cover learning-based state-of-the-art methods published after 2017. (*iv*) All traditional blur/ noise estimators base on prior assumptions that limit general application (e.g., the well-known dark channel prior only applies to scenes with dark spots and the homogeneity assumption to images without texture). On the other hand, it is worth to mention that the accuracy and robustness of learning-based estimators depend on a carefully selected training dataset to cover the target domain. Also note that NNs are still considered black-boxes and require means to assess their reliability [Bör+20]. (*v*) Learning-based models have been better studied as a basis for Physics-ML and Sensor AI, so we can draw on more matured approaches.

As learning-based blur estimator, we select the one of [Bau+18], since it is able to directly estimate a directional modulation transfer function (MTF) of an input image in

[1]The main time-varying noise sources of a camera system follow similar statistical distributions, which makes them inseparable when working with an image only. More details to the assumed noise models in Sec. 3.2.2.

order to objectively quantify image sharpness (details to MTFs in Sec. 3.3.1). Since the source code is not available, this approach must first be implemented. For comparison, we use [Bai+18] and [Wen+20] as two traditional state-of-the-art blur estimators that base on different prior assumptions (about image intensity and gradients, respectively) and whose source code is open available.

For learning-based noise estimation, we chose the one of [Tan+19], which provides fast pixel-wise estimations and the source code as well. We compare this estimator against the traditional state-of-the-art estimators [Shi+05] and [CZA15] as representatives of the two classes of spacial-based and transform-based noise estimators. The source code of [Shi+05] is not available as well and thus needs to be implemented.

Physics-ML/ Sensor AI

This thesis focuses on using physical knowledge about the sensor system to improve data-based parts of our proposed framework and not to improve the sensor system itself. Prior studies demonstrate several possibilities to combine both via (*i*) the loss function or initialization ("soft constraints"), (*ii*) altering the network design ("hard constraints"), or via (*iii*) hybrid methods. Each approach has advantages and drawbacks with respect to our camera self-health-maintenance scenario.

(*i*) Soft constraints are straightforward to implement, but naturally less effective than hard constraints. On the downside, the usage of a custom loss function or initialization only affects the training time, while an altered network design influences inference time. In our use case, an increase in training time is not considered significant and preferred over a longer inference time. We employ physical-models in order to generate the necessary amount of accurate ground truth data to train the data-based models. Thus, we inherently employ a form of physics-guided initialization in each model training.

When it comes to noise source estimation, we further investigate the most promising (*ii*) network design adaptation. More specifically, we input camera metadata along with a noisy image and let the network automatically learn how to fuse these heterogeneous data to solve the tasks of noise source identification and quantification. We do not expect longer inference times compared to classic noise estimation, since the respective calculations of the noise components can be represented with a few calculations in the network and the fusion of the sensor data on the decision-level is one possibility to be realized with low computation overhead.

In (*iii*) hybrid methods, both the physics-based and ML models are executed. Depending on physics-based model, calculations may not be performed in real time. This applies to extensive models in particular.

2.5 Summary

In this chapter, we provided an overview of studies that overlap with the different parts of our proposed framework.

Section 2.1 started with works on adaptive camera regulation, the task that combines online estimation of a camera's vision state and automatic execution of actions to improve some target criterion. This paradigm has been demonstrated for numerous applications of mobile machines and various readjusted camera parameters. Most of the works deal with object detection scenarios, camera exposure time and gain control, and learned image features. Only few works explicitly account for motion blur on a software basis, as it tends to be addressed with additional hardware elements (e.g., lens stabilizers).

In Sec. 2.2, we focused on blind blur and noise estimators to assess the quality of single images in order to determine a camera's vision state. The majority of traditional and learning-based blur estimators simultaneously estimate the kernel and the clean image on the basis of prior knowledge about image content (with focus on motion blur). Traditional noise estimators rely on assumptions in the spatial or frequency domain. Most noise estimators heavily rely on simplistic additive white Gaussian noise models and "*noise sources caused by digital camera electronics are still largely overlooked*" [Wei+20]. All learning-based approaches represent the blur/ noise estimation either explicitly, or, more frequently, implicitly within an end-to-end DNN. Recent studies tend to employ learning-based approaches over traditional ones.

In Sec. 2.3, we introduced the field of blending data-based methods with traditional scientific models as Physics-ML. If the focus lies on physics of a sensor system, the field is referred to as Sensor AI. Several authors investigated how a sensor system and AI approaches can benefit from each other. In the case of optical sensor systems, this can be done by learning sensor system component designs to optimize a particular task (e.g., free-form lens elements). On the other hand, AI can profit from contextual auxiliary data of a sensor system. Prior works further researched on how this sensor knowledge can be integrated into the AI and proposed physics-guided (*i*) loss functions (NN training loss extended by physics constraints), (*ii*) design and architecture (NN components tailored for a specific physics task), and (*iii*) hybrid modeling (parts of NN or physics are extended or replaced by each other).

In the end, we identified further goals for our self-health-maintenance framework from the advantages and disadvantages of related studies (Sec. 2.4). Specifically, these goals are:

(1) to optimize high-level image task performance against image quality for generality and interpretability,

(2) to consider motion blur as it occurs frequently with mobile machines and significantly affects subsequent high-level image processing,

(3) to examine image quality attributes in combination as they can influence each other,

(4) to determine and optimize camera parameters at runtime to account for dynamic scenes and camera systems,

(5) to utilize an extensive noise model for more realism in comparison to more frequent simple models,

(6) to use explicit and learning-based blur/ noise estimators as they favor independency from scenes and high-level tasks, interpretability, and real-time capability,

(7) and to employ physics-based ML initialization on physically generated data and a physical-guided ML design as it probably comes with low implementation complexity and computation time overhead.

CHAPTER 3

Theoretical Foundations

This theory chapter introduces the key concepts for understanding the complex processes within a camera to represent a scene in form of an image, how the targeted effects of blur and noise are caused, and how these effects are determined in a standardized way.

We first focus on the assumed camera systems and model their components: the sensor system and the lens system (Sec. 3.1). Subsequently, we introduce the notion of image quality to approach a camera's condition (Sec. 3.2) and present established methods to objectively assess image quality considering blur/noise (Sec. 3.3). Finally, we discuss on the model selections we made and corresponding limitations (Sec. 3.4), and summarize the key takeaway points (Sec. 3.5).

3.1 Camera System

A (digital) camera system denotes a device that captures a three-dimensional scene as a digital image. The development of the first digital camera dates back to 1978 [LS78] and decades of development have sophisticated even its basic structure (left of Fig. 3.1). Its components can be classified into three groups: (*i*) a lens system to direct and manipulate incoming light, (*ii*) a digital sensor system to sense and digitize the light, and (*iii*) auxiliary components. The (*i*) lens system can contain diverse lens types, one or many apertures, and light filters (e.g., ultraviolet (UV) filter and anti-reflection coatings); the (*ii*) sensor system a digital image sensor and processing electronics (e.g., for image post-processing), and the (*iii*) auxiliary components, for instance, an auto-focus unit and a temperature sensor.

To further reduce complexity, a camera system is simplified as depicted on the right of Fig. 3.1 to: a sensor system with the sensor and associated post-processing electronics (Sec. 3.1.1), and a lens system with a single lens and a single aperture (Sec. 3.1.2).

Figure 3.1: *Structure of a camera system.* Left: A typical camera system with an excerpt of its sophisticated setup: (1) various lenses, (2) multiple apertures, (3) diverse light filters, (4) a digital image sensor, (5) electronics for post-processing and (6) an auto-focus component. Right: Simplified camera system setup assumed in this work. Raw images based on [Dpr09; Kir09].

3.1.1 Sensor System

A (digital) sensor system can be defined as an electrical device that detects photons and converts them into digital signals. It typically consists of a two-dimensional array of photodiodes as photon detectors and electrical circuity to process the detector signals.

Camera sensors follow the same basic (idealized) principle of operation [KW14] (Fig. 3.2): first, a photodiode converts incoming photons into electrons during a configured time period (exposure time, t_{exp} [s]) and holds these electrons in a well, resulting in an electrical charge. A sense node transfers this electrical charge into measurable voltage

$$V = V_{\text{SN.Ref}} - (A_{\text{SN}} \cdot N_{e^-}), \tag{3.1}$$

with a reference voltage $V_{\text{SN.Ref}}$ [V], a gain A_{SN} [V/e$^-$], and the number of electrons N_{e^-}. This voltage V is then increased by a (source-follower) amplifier to

$$V' = A_{\text{SF}} \cdot V, \tag{3.2}$$

with a gain A_{SF} [V/V], to make it processable for an analog-to-digital converter, which quantizes this voltage into a digital number (DN) according to

$$\text{DN} = A_{\text{ADC}} \cdot (V_{\text{ADC.Ref}} - V'), \tag{3.3}$$

using a gain A_{ADC} [DN/V] and a reference voltage $V_{\text{ADC.Ref}}$ [V]. Doing this conversion for each photodiode, the resulting array of DNs forms the digital image, whereas a DN value is referred to as a picture element (pixel) intensity. In this work, it is assumed

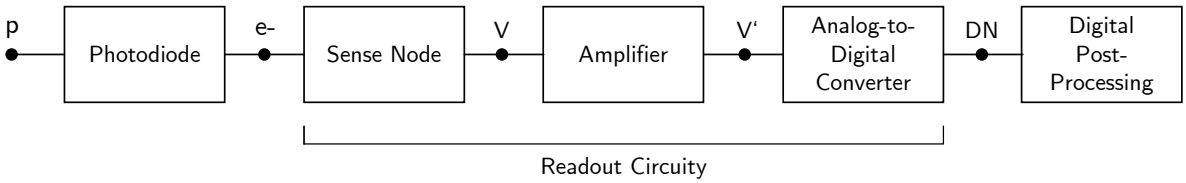


Figure 3.2: *Basic camera sensor processing pipeline.* A photodiode converts incoming photons (p) into electric charge (electrons, e^-) and a following sense node transforms this charge into voltage (V). The voltage gets amplified to V′ first before it passes an analog-to-digital converter that quantizes the voltage into digital numbers (DN). Typically, these DNs are post-processed on and/or off the sensor.

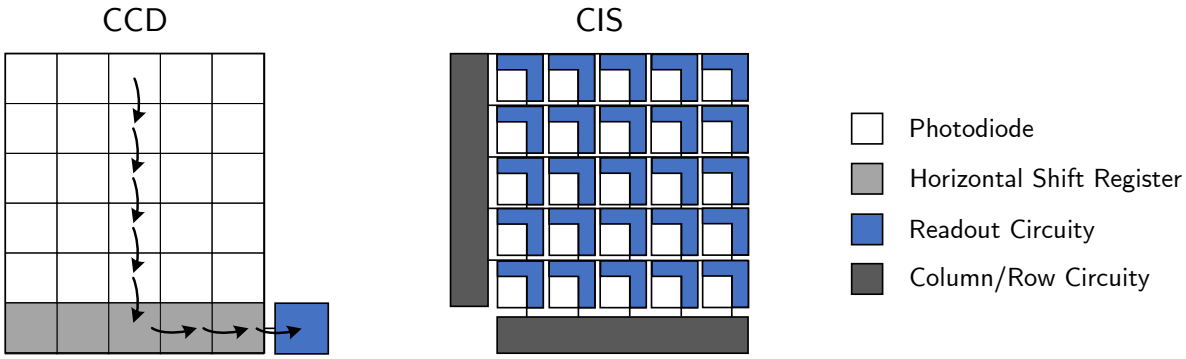


Figure 3.3: *Basic design of CIS and CCD sensors.* CCDs shift charge from each photodiode first vertically and then horizontally toward a single read out unit. In CISs, the charge is read out directly at the photodiode, addressed by column and row circuity, and further processed per row/column.

that the formed image is in a raw data format at this time[1]. This image is typically further post-processed by on- or off-chip located circuity [Wal13]. In the following, the composite of the sense node, the amplifier, and the analog-to-digital converter circuity is referred to as readout circuity [DH04, p. 197].

Nowadays, two types of camera sensors prevail the market: the charge-coupled device (CCD) and the active pixel sensor, known as a complementary metal–oxide–semiconductor (CMOS) imaging sensor (CIS). The CCD was invented in 1969 [CL69, p. 3] and the CIS around 1993 [Fos93]. Both sensor types differ in where the readout circuity is located (Fig. 3.3). A standard CCD comprises a single readout circuity unit at a corner of the sensor. As a consequence, each charge packet has to be moved toward this location. The arrows in Fig. 3.3 illustrate the trail of charge packet movements from a single column These "coupled" simultaneous shifts of charge packets yield the CCD its name. In contrast, a CIS comprises readout circuity per column, row, and photodiode for faster processing (however, the trend goes towards pixel circuity – see Sec. 3.4.1).

These different sensor architectures imply consequences related to transportation/ robotic applications and the camera condition monitoring of this work.

[1]Note that the raw data format is not standardized and that each camera manufacturer may already have applied a different processing.

The CCD's serial readout procedure introduces undesired correlations between pixels in form of artifacts like blooming, smear, and pixel defects that affect the readout of subsequent pixels in the serial readout [Jan01, p. 275, p. 406, ch. 5.4]. This serial readout is also the main reason for a higher power consumption of a CCD compared to a CIS, which makes it less practical for mobile devices [Wal13]. On the other hand, the CIS technology heavily benefits from advances of general CMOS manufacturing in several ways. Standardized manufacturing steps and the constantly reduced size of circuity reduces the fabrication cost and enable on-chip (and in-pixel) processing that encourage sensor AI [GC16]. Nowadays, standard on-chip processing may include global shutter, noise suppression, pixelwise analog-to-digital conversion [HL11, ch. 4.3]. However, additional electronic components come at the expense of a reduced photodiode size (limiting the light sensitive area and the well capacity to hold electrons and thus the dynamic range, i.e., the imageable amount of different light intensities) [Wal13]. To this end, microlenses can focus incoming light to the remaining light sensitive area. A more advanced technique is the fabrication of backside-illuminated CISs that have their photodiodes on the backside and the readout electronics on the front [Jan01, sec. 3.4]. However, this is still considered cost-intensive compared to frontside-illuminated sensors [Vic+20]. In the end, "there is no such thing as a free lunch" [Fri75], so the sensor setup should be chosen with the requirements of the intended target application in mind.

The predominant fields of application base on the strengths of the respective sensor type. Currently, CCDs are preferred in medial and scientific applications that demand high dynamic ranges and low noise over other factors (e.g., in digital radiography) [Wal13; Que+20]. When it comes to high frame rates, low power consumption, compact sizes, a limited budget, or on-chip processing, usually the CIS is the first choice [GC16]. From a historical point of view, engineers consider the CIS as the successor of the CCD since the CIS surpasses the CCD in an increasing number of industrial and scientific use cases with respect to their specific demands [GC16; Fos20].

3.1.2 Lens System

In this work, a lens system is simplified to a combination of a lens and an aperture. The lens serves to project the three-dimensional environment onto the two-dimensional sensor plane, while the aperture controls the amount of passing light.

The first usage of a lens within a camera can be dated back to the mid of the 16th century [Ila07, p. 219] and resulted from the limitations of a lensless camera, which can be modeled by the well known pinhole camera model (named after its tiny aperture) [Kin92, pp. 63–64]. In this model, the aperture not only controls the incident amount of light, but also the sharpness of a projected object point from the environment. Increasing the aperture diameter (D_A [mm]) would likewise increase the projected point size (D_I [mm]), as the aperture simply creates a cone of incoming light rays from an object point (cf. left half of Fig. 3.4).

On the other hand, decreasing D_A would lead to darker images and D_I is lower-bounded by diffraction. Darker images are typically tackled by longer t_{exp} resulting in motion blur from moving objects (see Sec. 3.2.1.2).

In contrast to the pinhole model, the use of a lens removes this dependency from D_I to D_A, since the lens is able to focus the incident cone of light rays onto a point (in the ideal case, see right half of Fig. 3.4). As a consequence, a lensed camera system can produce brighter images using a D_A larger than pinhole-size without increasing D_I. The higher light exposure leads to a shorter t_{exp} required that also reduces the potential motion blur. But these advantages come at the expense of two new limitations: (i) for D_A larger than pinhole-size, D_I now depends on the distances from the lens to the object point (d_O [mm]) and to the sensor plane (d_I [mm]) (details in Sec. 3.2.1.1), and (ii) the lens introduces projection artifacts known as lens aberrations [Kin92, pp. 37–48]. Lens aberrations are typically mitigated to a negligible level for many applications by incorporating additional lenses, lens coatings, additional apertures placed before or after lenses, and an offline camera calibration before the field usage. Hence, the effect of lens aberrations is neglected in the following.

This work follows the thin lens model [Hec17, pp. 165–168] with a single perfect thin and convex lens, a single round aperture, and light modeled as rays (Fig. 3.4) – however, this model is still frequently used in the literature to describe the light geometry of more complex lens systems [FL19; Seo20]. It models the projection of object points along the optical axis of the lens onto the sensor plane by the Gaussian Lens Formula

$$\frac{1}{f} = \frac{1}{d_O} + \frac{1}{d_I}, \tag{3.4}$$

that relates the focal length (f) of the lens to its distances to the projected object point (d_O) and to the sensor plane (d_I). For further simplification, the aperture is considered to control the amount of incoming light only, while an electronic global shutter regulates the time of light exposure t_{exp}.

3.2 Image Quality

We utilize the quality of an image that is produced by a camera system as a mean to infer the camera's condition, since many observable changes in image quality are direct effects of camera system processes. There is no unified definition for image quality[2], but multiple definition approaches [JB97; Pet00; BPA02; Kee02; Eng04; WBS09], [PE18, p. 29].

[2] In fact, the term image quality is often used undefined in standard work as if it was part of common knowledge (e.g., in [Jah00; JEV09; Dav12; Kle14]).

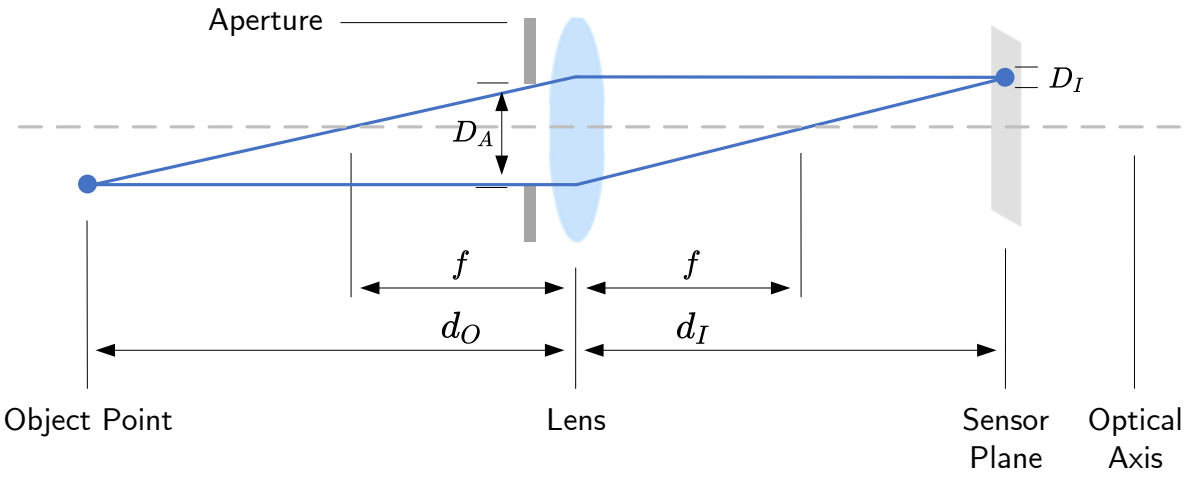


Figure 3.4: *Thin lens model.* A single perfect, thin and convex lens projects an object point along the optical axis onto the sensor plane. The projection's size (D_I) is determined by the focal length (f) of the lens and the lens' distances to the object point (d_O) and to the sensor plane (d_I), respectively. The aperture of size D_A controls the amount of light.

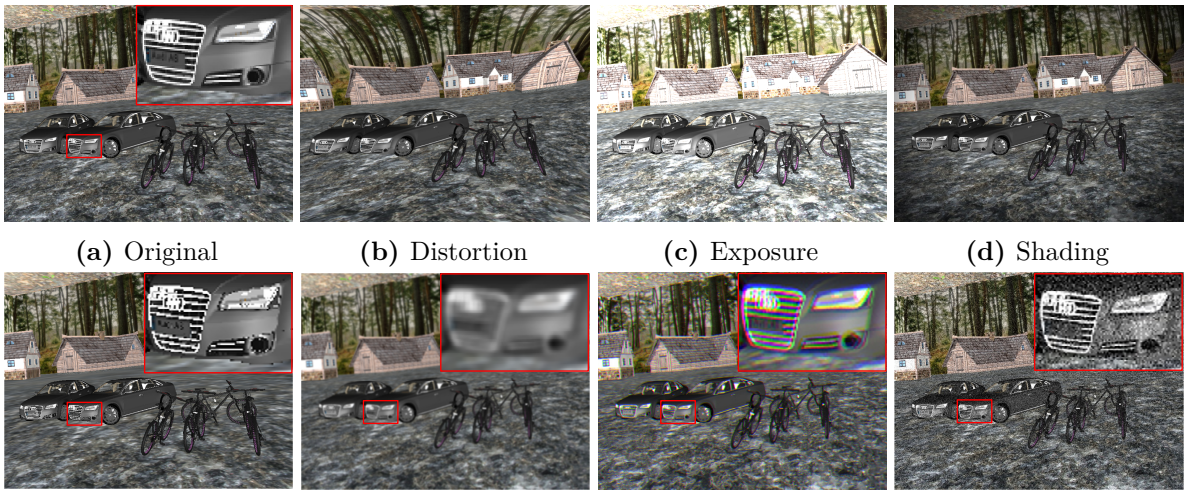
We follow the definition from [BPA02],

> "[Image quality is] the weighted combination of all of the visually significant attributes of the image, when considered in its intended marketplace or application.",

as it comprises two key aspects that several definitions address: multiple visual image attributes contribute to image quality and their respective weighting depends on the intended application of an image (e.g., object detection, see Sec. 1.1).

Phillips and Eliasson categorize image attributes into global and local ones, whereas global ones are considered essentially independent to magnification and changing viewing distances, in comparison to local ones [PE18, p. 35]. Global attributes include, for instance, exposure, optical distortions, and shading. Local attributes contain blur, color fringing, noise, and different artifacts. Figure 3.5 depicts exemplary image attributes that typically originate in a camera system. Note that it depicts only a small part of the possible attributes and each of them can be further sub-classified according to their respective root causes. We focus on the local attributes blur and noise, as two of the most common effects researched in the literature (see Sec. 1.1).

Let us further emphazise the dependence of an image target application on image quality using image blur as an example (Fig. 3.6). Blur is considered an image degradation when it comes to human interaction and many Computer Vision applications (e.g., edge detection) [NPJ83; NHM17]. However, blur can also be beneficial for modern learning-based object detection approaches (see Sec. 7.1) and for navigation by means of star tracking (to locate stars in sub-pixel accuracy) [Lie95]. That is, blur and other image attributes should only be judged in the context of a desired target application. To this end, the sensitivity of image attributes for a respective target application must be investigated in order to assess a camera's condition.



(a) Original **(b)** Distortion **(c)** Exposure **(d)** Shading

(e) Artifacts **(f) Blur** **(g)** Color Fringing **(h) Noise**

Figure 3.5: *Exemplary image quality attributes.* Top row: (a) original image and global attributes including (b) distortion (e.g., pincushion distortion), (c) exposure (e.g., overexposure), and (d) shading (e.g., vignetting). Bottom row: local attributes including (e) artifacts (e.g., aliasing), (f) blur (e.g., defocus blur), (g) color fringing (e.g., chromatic aberration), and (f) noise (e.g., readout noise). This work is limited to blur and noise only.

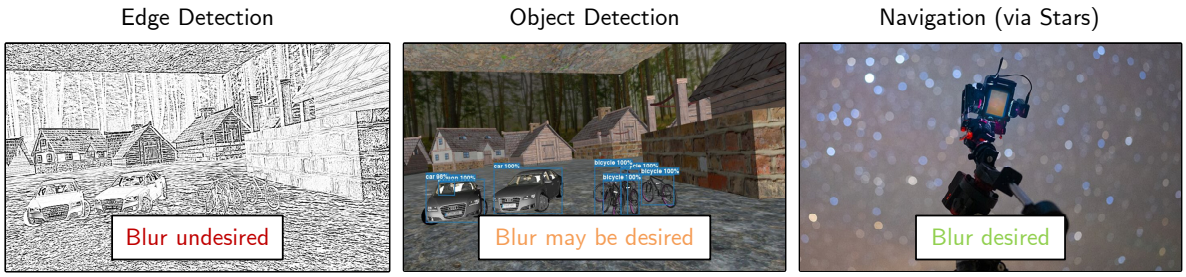


Figure 3.6: *Dependence of target applications on image quality on the example of blur.* There are applications where blur is undesirable (e.g., edge detection), where blur may be beneficial (see Sec. 7.1) or where blur is desirable (e.g., star tracking for navigation; image from [Zaf23]).

The image formation process in terms of blur and noise that we consider in a camera is illustrated in Fig. 3.7. The following sections present blur and noise in a top-down approach, from the visible image effect toward the respective root causes linked to the camera system, and means to model these processes (Secs. 3.2.1 and 3.2.2).

3.2.1 Blur

Image blur denotes the result of processes that reduce image sharpness. The most prominent of such processes are (*i*) light refracted by a defocused lens system, (*ii*) relative motion between the sensor and the scene, (*iii*) atmospheric turbulence, and (*iv*) diffraction [JEV09, p. 325]. Processes (*iii*) and (*iv*) cannot be avoided from a camera's point of view, so we focus on the former two sources, whose induced blur types are called defocus and motion blur, respectively (Fig. 3.8). Many factors contribute to these processes and make their mathematical description complex.

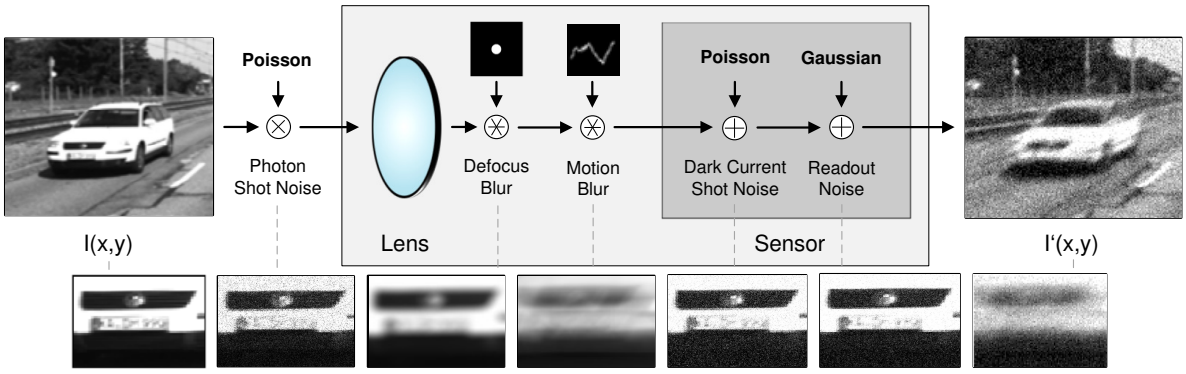


Figure 3.7: *Image formation process* of the considered camera system, including blur and noise models. A clean image $I(x, y)$ undergoes several physical processes that produce noise and blur, yielding the corrupted image $I'(x, y)$ (clean image patch vs. distinct corruptions in stated order). Noise is either signal-dependent or signal-independent, while blur is modeled as a convolution with a point spread function (PSF).

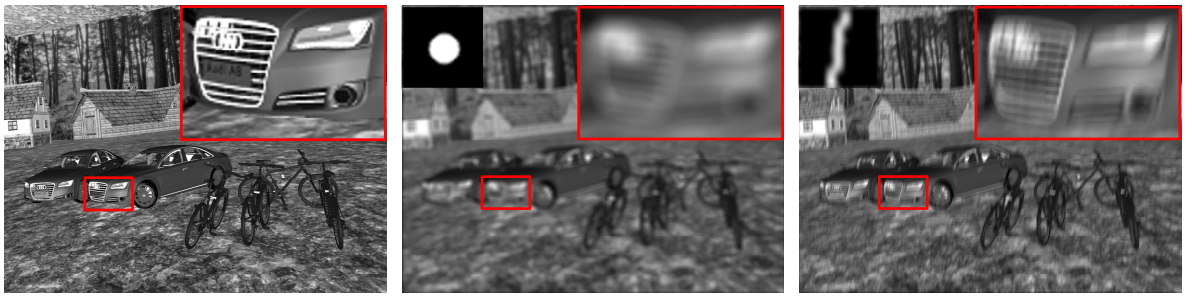


Figure 3.8: *Blur sources comparison* between the original scene (left), simulated defocus blur (middle), and simulated motion blur (right) with their respective blur kernels $h(x, y)$ in the top-left corners. Defocus blur is characterized by isotropy, while motion blur kernels are typically directional.

For the sake of simplicity, they are often modeled as a convolution on the image plane:

$$I^*(x, y) = I(x, y) \circledast h(x, y), \tag{3.5}$$

where $I \colon \mathbb{N}^2 \to [0, 255]$ is the input intensity at pixel (x, y) before the blur process, $h \colon \mathbb{N}^2 \to [0, 255]$ the blur kernel, and $I^* \colon \mathbb{N}^2 \to [0, 255]$ the blurred image intensity. The kernel h is also called point spread function (PSF) [JEV09, p. 328]. Let us now describe defocus and motion blur kernels h.

3.2.1.1 Defocus Blur

We model a defocus blur kernel $h(x, y)$ to distribute a pixel's intensity evenly over an approximate circular area of neighboring pixels (with radius $r \in \mathbb{R}$ and center $(c_x, c_y) \in \mathbb{R}^2$):

$$h(x, y) = \begin{cases} s, & (x - c_x)^2 + (y - c_y)^2 \leq r^2 \\ 0, & \text{otherwise,} \end{cases} \tag{3.6}$$

with the value $s \in \mathbb{R}$ determined by the normalization constraint $\iint h(x,y)\,dx\,dy = 1$ [JEV09, p. 325]. This circle corresponds to the object point projected through the lens onto the sensor plane (known from Fig. 3.4) and refers to the term circle of confusion (CoC). Building on top of the light geometry of the assumed thin lens model (Sec. 3.1.2), the CoC's diameter $D_I = 2rd_p + 1$ can be calculated as

$$D_I = D_A \frac{f}{d_O - f} \frac{|d_B - d_O|}{d_B} \quad \text{with} \quad \frac{1}{f} = \frac{1}{d_O} + \frac{1}{d_I}, \tag{3.7}$$

expressed in terms of the focused object distance (d_O), the out-of-focus object distance (d_B), the focused image distance (d_I), the focal length (f), and the aperture diameter (D_A) [Ray02, p. 216]. Note that the sensor's pixel pitch d_p [mm] transforms the radius r from the image space to the metric space (assuming contiguous square pixels).

A projected object point is considered defocused if its CoC is larger than d_P, i.e., $D_I > d_P$. In theory, there is a tolerance value range $[a, b]$ for each involved parameter to ensure focused projections:

$$\forall p, q \in \{f, D_A, d_O, d_B, d_I\},\ \exists [a,b] \subset \mathbb{R},\ (p \in [a,b] \wedge D_I \leq d_p)_{p \neq q}, \tag{3.8}$$

with fixed parameters $p \neq q$ for simplicity. The possibility to control these parameters allows us to avoid or counteract defocus. In practice, d_O and d_B are the least controllable but most experienced parameters during field operation, so their associated tolerance range is most regarded in literature, known as depth of field (DoF, Fig. 3.9) [Kin92, ch. 5]. As a consequence, fixed-focus cameras are typically focused on the so called hyperfocal distance H [mm] that maximizes the DoF to $[H/2, \infty]$ [Kin92, p. 89]:

$$H = D_A \frac{f}{d_p} + f. \tag{3.9}$$

In the following, we assume our camera systems to be focused on H with an operational distance of $[H/2, \infty]$. We further assume an image-patch-wise constant distance of the depicted object scene to the camera (as we consider image-patch-wise/ spatially-varying monitoring, cf. Sec. 1.1). Both assumptions imply a negligible contribution of d_O and d_B to potential defocus blur, and thus support the model chosen in (3.6).

3.2.1.2 Motion Blur

Motion blur emerges due to relative motion between the camera and the recorded object scene during the exposure time t_{exp}. Depending on the type of motion, motion blur can manifest as translation, rotation, scale changes or a combination of all of them. Hence, a closed-form expression for $h(x,y)$ may be complex to obtain. Jayaraman et al. [JEV09, pp. 325–326] exemplarly express $h(x,y)$ for a simplified translational motion with a constant velocity v_{rel} [px/s] under an angle of Φ [rad] to the horizontal axis of the

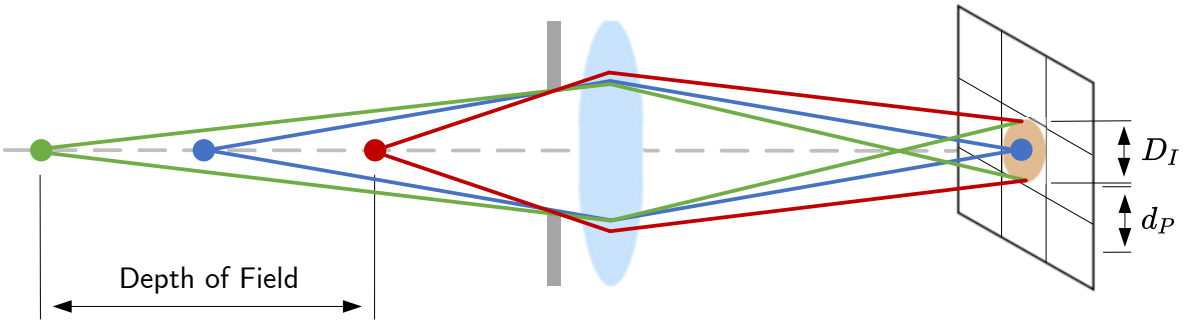


Figure 3.9: *Depth of field* describes the tolerance range of object distances to the lens that do not induce a defocus blurred projection on the image (remaining parameters fixed), i.e., whose circle of confusion diameter (D_I) is lower or equal to a pixel's pitch (d_P).

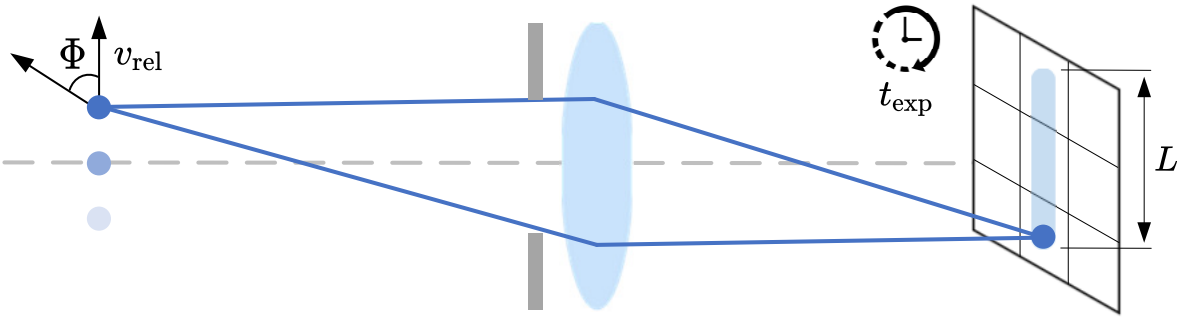


Figure 3.10: *Motion blur* results from relative motion between an imaged object point and the camera during exposure. The length L, the shape, and the pixelwise intensities of the blur path depend on the relative motion velocity v_{rel} and the relative motion angle Φ (horizontal to the sensor) at each time instant during the exposure interval $[0,\, t_{\text{exp}}]$.

camera sensor as (Fig. 3.10):

$$h(x,y) = \begin{cases} \frac{1}{L}, & \sqrt{x^2+y^2} \leq \frac{L}{2} \text{ and } \frac{x}{y} = -\tan(\Phi) \\ 0, & \text{otherwise} \end{cases} \quad \text{with} \quad L = v_{\text{rel}} \cdot t_{\text{exp}}. \tag{3.10}$$

In reality, both Φ and v_{rel} may not be constant during the exposure interval, e.g., because of factors like an uneven driving ground or unpredictable moving scene objects. Hence, we consider simplified linear as well as complex non-linear movements to model $h(x,y)$. In general, we constrain a motion blur kernel $h(x,y)$ to contain a coherent path of pixels with non-zero and potentially inhomogeneous intensities (cf. right kernel of Fig. 3.8).

Analogous to defocus blur, we consider a projected object point to be motion blurred if $L > d_p$, and can likewise derive tolerance ranges for the involved parameters to avoid blur:

$$\forall p, q \in \{t_{\text{exp}}, v_{\text{rel}}, \Phi\},\ \exists [a,b] \subset \mathbb{R},\ (p \in [a,b] \wedge L \leq d_p)_{p \neq q}, \tag{3.11}$$

with fixed parameters $p \neq q$. In transportation and robotic applications, v_{rel} and Φ lie beyond the operational range of a camera system and are controlled by superordinated tasks (e.g., save movement through the environment); so only t_{exp} remains to control motion blur.

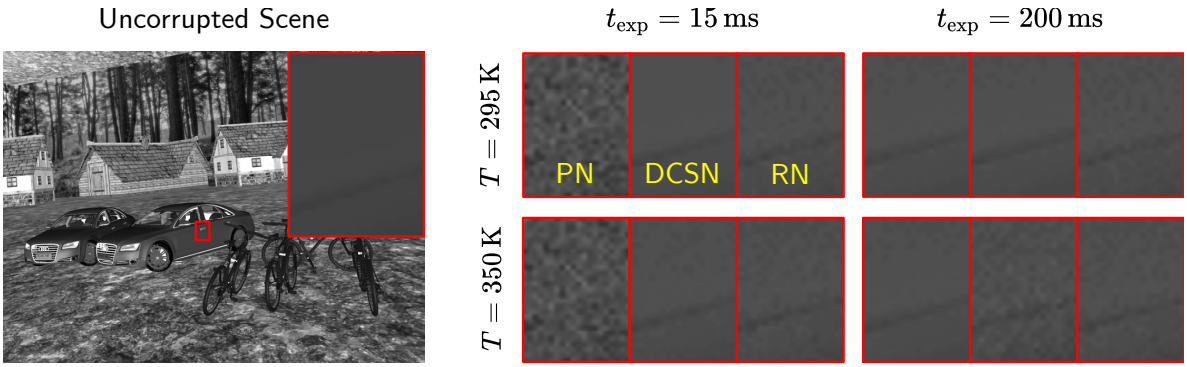


Figure 3.11: *Comparison of noise sources.* Left: Uncorrupted scene. Right: Scene with simulated photon shot noise (PN), dark current shot noise (DCSN), and readout noise (RN) for different exposure times (t_{exp}) and temperatures (T). PN depends only on t_{exp}. When t_{exp} increases, PN becomes less noticeable as the signal grows faster. DCSN depends on both t_{exp} and T, with a greater influence of T. RN depends on T only. For the simulation, parameters of the Prosilica GC1380H camera system were used (see Tab. A.2).

3.2.2 Noise

Image noise denotes "any undesired information that contaminates an image" [JEV09, p. 348] and may originate from the environment, the camera system, during transmission, and from processing steps. Having the online condition monitoring approach in mind, we limit this study to camera system noise, but still, each involved camera component can introduce noise due to physical imperfection or inaccuracies [BJ15] – however, most noise originate within the image sensor [PE18, p. 207]. In addition, we consider only time-varying sources because time-invariant noise sources (such as photo response non-uniformity) are often addressed during calibration (before acquisition) and their residuals are assumed to have a minor influence on image quality. Following this scope reduction, the remaining major noise sources can be limited to (*i*) photon shot noise, (*ii*) dark current shot noise, and (*iii*) noise from readout circuity (readout noise) [PE18, p. 83], [Jan01, pp. 101–102][3]. Figure 3.11 visualizes the noise sources (*i*) – (*iii*).

Generally, noise can be modeled by:

$$\widetilde{I}(x,y) = I(x,y) + I(x,y)^{\gamma}\, u(x,y), \tag{3.12}$$

where $I\colon \mathbb{N}^2 \to [0,255]$ is the clean intensity (the signal's intensity), $u\colon \mathbb{N}^2 \to [0,255]$ is a random, stationary and uncorrelated noise process, and $\widetilde{I}\colon \mathbb{N}^2 \to [0,255]$ is the noisy intensity. A parameter $\gamma \in \mathbb{R}$ controls different noise types. The amount of noise (or noise level) may be quantified using the standard deviation σ of the underlying statistical distribution of u. The following sections detail the noise sources (*i*) – (*iii*). As a theoretical guide, we follow [KW14].

[3] Janesick defines readout noise as a collection of "noise sources that are independent of the signal level" and hence counts dark current shot noise as a part of readout noise [Jan01, p. 101]. We follow another definition approach to distinguish between these two types of noise (see Section 3.2.2.3).

3.2.2.1 Photon Shot Noise

Photon shot noise is based on two aspects: the generation of photons from a light source underlies random fluctuations and, as a consequence, they arrive independently at the photodiode within the exposure time interval. Although the origin is only loosely coupled to the camera system, we still consider this one here because of its non-negligible impact on image quality [Jan01, pp. 101–102].

Photon shot noise follows a Poisson distribution $\mathcal{P}\colon (\mathbb{R} \to \mathbb{R}) \to \mathbb{R}$. If the expected number ($\lambda = \sigma^2$) of arriving photons (x) is large enough (i.e., in non-low illumination conditions), the Poisson distribution may be approximated by a Gaussian distribution $\mathcal{N}\colon (\mathbb{R}^2 \to \mathbb{R}) \to \mathbb{R}$ using the Central Limit Theorem [Dev11, p. 225]:

$$\mathcal{P}(\lambda)(x) = \frac{\lambda^x}{x!} e^{-\lambda} \overset{x \to \infty}{\approx} \frac{1}{\sqrt{2\pi\lambda}} e^{-(x-\lambda)^2/2\lambda} = \mathcal{N}(\lambda, \lambda)(x). \tag{3.13}$$

The higher the number of arriving photons, the higher the number of random fluctuations; hence photon shot noise behaves signal-dependent and can be described by (3.12) when setting $\gamma = 1$ and $u \sim P(\lambda)(x)$. Please note that a reduction of the noise level σ can only be achieved by a lowered signal x.

3.2.2.2 Dark Current Shot Noise

Similar to photon shot noise, dark current shot noise (DCSN) originates from the random arrival of dark current (DC) electrons and follows the same distribution (3.13). DC emerges from thermally generated electrons at different sensor material regions [Jan01, sec. 7.1.1]. Its expected average $\widehat{\mathrm{DC}}$ [e$^-$/sec/px]$\colon \mathbb{R} \to \mathbb{R}$ can be modeled as

$$\begin{aligned} \widehat{\mathrm{DC}}(T) &= 2.5 \times 10^{15} d_p^2 D_{\mathrm{FM}} T^{1.5} \exp\left(-\frac{E_{\mathrm{g}}(T)}{2kT}\right), \\ E_{\mathrm{g}}(T) &= E_{\mathrm{g0}} - \frac{\alpha T^2}{\beta + T}, \end{aligned} \tag{3.14}$$

with the pixel area d_p^2 [cm^2], the dark current figure-of-merit D_{FM} [nA/cm] at 300 K, the temperature dependent band gap energy of the semiconductor E_g [eV]$\colon \mathbb{R} \to \mathbb{R}$, the temperature T [K], the Bolzmann's constant k, and the material depending terms E_{g0} [eV], α, and β ($\widehat{\mathrm{DC}}$ from [Jan01, p. 622], E_{g} from [Pan75, p. 27], material specific terms for silicon in [KW14]). Integrating $\widehat{\mathrm{DC}}$ over the exposure time t_{exp} leads to the overall dark signal $S_{\widehat{\mathrm{DC}}}$ [e$^-$/px]. The DCSN is then determined by its noise level σ_{DCSN} [rms e$^-$] with

$$\sigma_{\mathrm{DCSN}} = \sqrt{S_{\widehat{\mathrm{DC}}}} = \sqrt{\widehat{\mathrm{DC}} \cdot t_{\mathrm{exp}}}, \tag{3.15}$$

and follows (3.13) using $\sigma^2_{\mathrm{DCSN}} = \lambda$ [Jan01, p. 626]. DCSN behaves signal-independent, hence $\gamma = 0$ in (3.12) [Jan01, ch. 7.1.1]. At runtime, DCSN can be controlled via T or t_{exp}.

3.2.2.3 Readout Noise

Readout noise refers to the imperfections due to the sensor's electronic circuitry converting charge into digital values and it is attributed to the on-chip amplification and conversion processing units [DH04, p. 197]. Although readout noise can be reduced to a low level in scientific cameras [Jan01, p. 36], its impact may be still significant for industry-grade sensors that lack noise reduction [Fos20]. We incorporate sense node reset noise and source-follower (amplifier) noise as the main time-varying components, whereas source-follower noise can further be sub-divided into Johnson-Nyquist noise, flicker noise, and random telegraph noise. We refer to the original sources [Jan07, sec. 11] and [KW14] for details to (3.16) – (3.19) and for default values.

Sense node reset noise (alias kTC noise) results from thermal noise by the channel resistance of the reset transistor that periodically resets the sense node to a reference level for charge sensing [Jan01, p. 537–538]. Its noise level σ_{SN} [rms e^-] follows

$$\sigma_{\mathrm{SN}} = \frac{\sqrt{kTC}}{q}, \tag{3.16}$$

with the Bolzmann's constant k, the temperature T [K], the sense node capacitance C [F], and the electric charge q [C]. A signal processing technique called correlated double sampling (CDS) typically eliminates sense node reset noise for CCDs, but may increase thermal noise in CISs due to their pixelwise CDS implementation ([Dep+00]). Sense node reset noise can be described by (3.12) when setting $\gamma = 0$ and $u \sim \mathcal{N}(0, \sigma_{\mathrm{SN}}^2)$, and can be regulated with T.

Let us focus on the source-follower noise components. Johnson-Nyquist noise originates from the source-follower's resistance that induces erratic motion of electrons in the current (measured as thermal noise, similar to sense node noise). It is also referred to as white noise, as its magnitude is independent of the frequency in the power spectrum. In contrast to Johnson-Nyquist noise, flicker noise (also $1/f$ noise) varies approximately with the inverse of the frequency and results from imperfect contracts between two materials and tunneling of electrons into the oxide [Jan01, p. 544]. CISs are affected from higher $1/f$ noise than CCDs as they contain circuity in each pixel. Random telegraph noise arises from random trapping and emission of electrons, and leads to discrete modulation of the channel current [TE00]. The source-follower noise level σ_{SF} [rms e^-] can be approximated as

$$\sigma_{\mathrm{SF}} \approx \frac{\sqrt{\sum_{f=1}^{f_{\mathrm{clock}}} S_{\mathrm{SF}}(f) H_{\mathrm{CDS}}(f)}}{A_{\mathrm{SN}} A_{\mathrm{SF}} \left(1 - \exp\left(-t_s/\tau_D\right)\right)}, \tag{3.17}$$

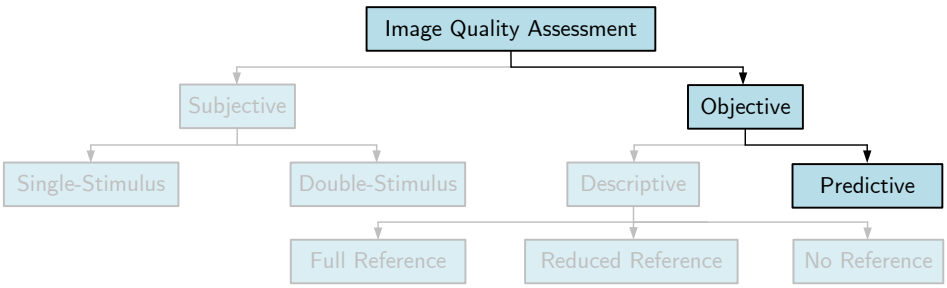


Figure 3.12: *Classification of image quality assessment approaches.* This thesis focuses on methods that are independent of human observers (objective) and that predict individual image quality attributes (predictive). Inspired by [TR09].

with the readout frequency f_{clock} [Hz], the sense node conversion gain A_{SN}, the source-follower gain A_{SF}, the CDS sample-to-sample time t_s [sec], and the CDS dominant time constant τ_D [sec]. The source-follower noise power spectrum $S_{\text{SF}}\colon \mathbb{R}^2 \to \mathbb{R}$ may be modeled by

$$S_{\text{SF}}(f,T) = W^2(f,T)\cdot\left(1+\frac{f_c}{f}\right)+S_{\text{RTN}}(f) \quad \text{with} \quad S_{\text{RTN}}(f) = \frac{2\Delta I^2\tau_{\text{RTN}}}{4+(2\pi f\tau_{\text{RTN}})^2}, \tag{3.18}$$

the white noise power spectrum W [rms V/Hz]$\colon \mathbb{R}^2 \to \mathbb{R}$, the flicker noise corner frequency f_c [Hz], and the random telegraph noise power spectrum $S_{\text{RTN}}\colon \mathbb{R} \to \mathbb{R}$ with its characteristic time constant τ_{RTN} [sec] and the induced source-follower current modulation ΔI [A]. The CDS transfer function $H_{\text{CDS}}\colon \mathbb{R} \to \mathbb{R}$ follows

$$H_{\text{CDS}}(f) = \left(\frac{1}{1+(2\pi f\tau_D)^2}\right)\cdot(2-2\cos(2\pi f t_s)). \tag{3.19}$$

Source-follower noise can be described by (3.12) when setting $\gamma = 0$ and $u \sim \mathcal{N}(0,\sigma_{\text{SF}}^2)$. In theory, it can be controlled by $A_{\text{SN}}, A_{\text{SF}}, T, f_{\text{clock}}, t_s$, and τ_D. In practice, only the gains and the temperature are accessible. CDS may also be employed to tackle the Johnson-Nyquist and flicker noise contributions [Jan01, p. 556].

3.3 Image Quality Assessment

Image quality can be assessed with subjective or objective methods (Fig. 3.12). Subjective methods score image quality based on a aggregation of human opinions. Depending on whether only a test image is given or in combination with a source image, it can be further classified as single- or double stimulus, respectively [WB22, pp. 1–13].

Objective image quality assessment (IQA) relies on metrics independent of human perception and can be descriptive (i.e., they describe the quality of a specific image) or perceptive (i.e., they predict image quality of any image made by a certain camera in terms of image quality attributes) [PE18, p. 32]. Analogously to subjective methods,

descriptive approaches can be further separated based on whether a reference to a test image is given or not (full reference: reference image given; reduced reference: features of reference image given; no reference: only test image given). This thesis focuses on methods that are to be evaluated with regard to specific image target application performances (objective) on the basis of individual image quality attributes to assess the quality of images from a specific camera (predictive)[4].

In the following sections, standard methods to estimate the chosen image quality attributes blur (Sec. 3.3.1) and noise (Sec. 3.3.2) are presented.

3.3.1 Blur Estimation

The PSF: $\mathbb{N}^2 \to \mathbb{R}$ of a blur process $h(x, y)$ (3.5) can be used to objectively quantify image blur [JEV09, p. 328]. Its Fourier transform is the optical transfer function (OTF: $\mathbb{R} \to \mathbb{R}$) and it describes how spatial frequencies f (i.e., image details, contrast) are affected by blur:

$$\mathrm{PSF}(x, y) \quad \overset{\mathcal{F}}{\mapsto} \quad \mathrm{OTF}(f) \propto \mathrm{MTF}(f)\, e^{i\,\mathrm{PhTF}(f)}. \tag{3.20}$$

Usually only the magnitude of the OTF, known as the modulation transfer function (MTF: $\mathbb{R} \to \mathbb{R}$), is relevant to quantify blur, and so the phase transfer function (PhTF: $\mathbb{R} \to \mathbb{R}$) is omitted [Hec17, p. 580]. To interpret MTF values easier, we use only normalized values $\mathrm{MTF}_{\mathrm{norm}}\colon [0, 1] \to [0, 1]$ and $f_{\mathrm{norm}}\colon \mathbb{R} \to [0, 1]$ in terms of pixel units (i.e., in the image space) in this thesis:

$$\mathrm{MTF}_{\mathrm{norm}}(f_{\mathrm{norm}}) \doteq \frac{MTF(f_{\mathrm{norm}})}{C_0}, \qquad f_{\mathrm{norm}}(f) \doteq \frac{f}{f_{\mathrm{Nyquist}}}, \tag{3.21}$$

with the Nyquist frequency f_{Nyquist} [Jan07, p. 128] and the maximum possible contrast modulation C_0 at $f = 0$. This normalization can be reversed to compare MTF values among different camera systems.

In the following, we introduce two standard approaches to determine an MTF that either characterize one or two image dimensions: the slanted-edge method (Sec. 3.3.1.1) and the Siemens star method (Sec. 3.3.1.2). For both methods, we use the "resolving power tool" developed by Meißner [Mei20].

3.3.1.1 Slanted-Edge Method

The slanted-edge (SLE) method [Sta17; Bur+00] was first published in the ISO standard 12233 and can approximate the MTF along one image direction. Let us introduce the

[4]The author's use of the term "predictive" is consistent with the intent of what we refer to as assessment. Strictly speaking, the predictive paradigm is a bottom-up approach and ours a top-down approach, however, both are comparable.

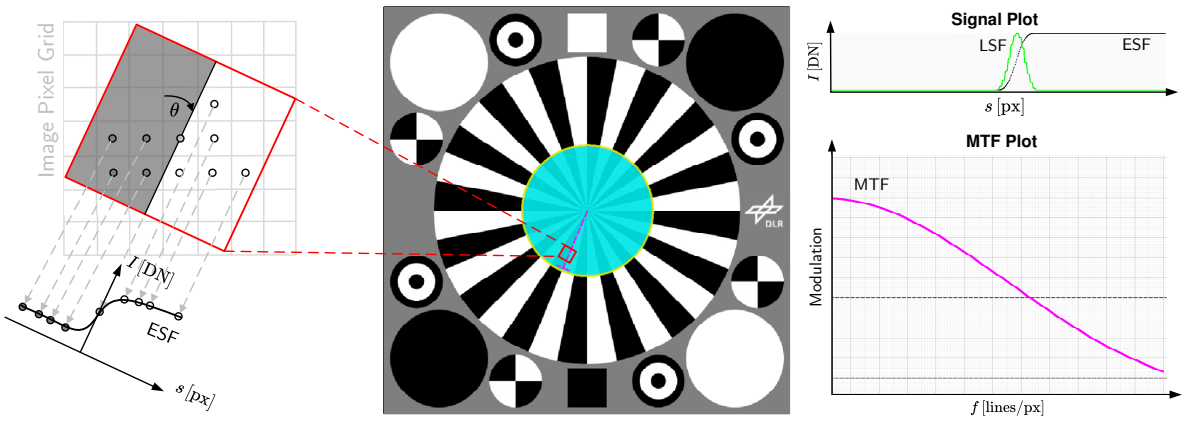


Figure 3.13: *Slanted-edge method for one-dimensional blur estimation.* Exemplary application on a Siemens star target image. The pixel intensities neighboring the slanted edge (SLE) of interest with an inclination angle θ are first projected onto a perpendicular vector to form the edge spread function (ESF). The magnitude of the ESF's first derivative (called line spread function, LSF) translated into the Fourier space leads to the modulation transfer function (MTF). Left illustration inspired by [Van19].

procedure on the example of the horizontal image direction (Fig. 3.13): According to the standard, the method requires an arbitrary slanted knife-edge with an edge contrast of $C = 4:1$ and an inclination angle of $\theta = 5\,^{\circ}$. First, a region of interest around the edge is extracted either manually or automatically (middle part of Fig. 3.13). The intensity values of pixels along the edge and neighboring the edge are then projected onto a vector perpendicular to the edge. In this step, θ enables to sample the edge intensities in sub-pixel spatial resolution and C ensures a suitable radiometric resolution (left part of Fig. 3.13). Fitting these projected intensities forms the edge spread function (ESF), which describes the camera response to the edge. The first derivative of the ESF then yields the line spread function (LSF), i.e., the camera response to the line. Lastly, the absolute value of the Fourier transformed LSF yields the MTF (right part of Fig. 3.13).

3.3.1.2 Siemens Star Method

The Siemens star method [Reu+04; Reu+06; Mei20] approximates the MTF in two image dimensions. It relies on imaging a Siemens star target [Sta22], which is a circular pattern that consists of a number of f_s alternating black-white segments, whose spatial alternation frequency $f\,[px^{-1}]$ increases towards the Siemens star center (Fig. 3.14). The frequency f is calculated as

$$f = \frac{f_s}{\pi r}, \tag{3.22}$$

with a scan radius $r\,[px]$ (see exemplary yellow circle in Fig. 3.14). We follow the ISO standard 15775 [Sta22] to use $f_s = 32$ and an edge contrast of $C = 4:1$. For every scanned r, one can calculate the discrete contrast transfer function $\mathrm{CTF}_D\colon [0,1] \to [0,1]$ as

$$\mathrm{CTF}_D(f) \doteq \frac{I_{\max}(f) - I_{\min}(f)}{I_{\max}(f) + I_{\min}(f)}. \tag{3.23}$$

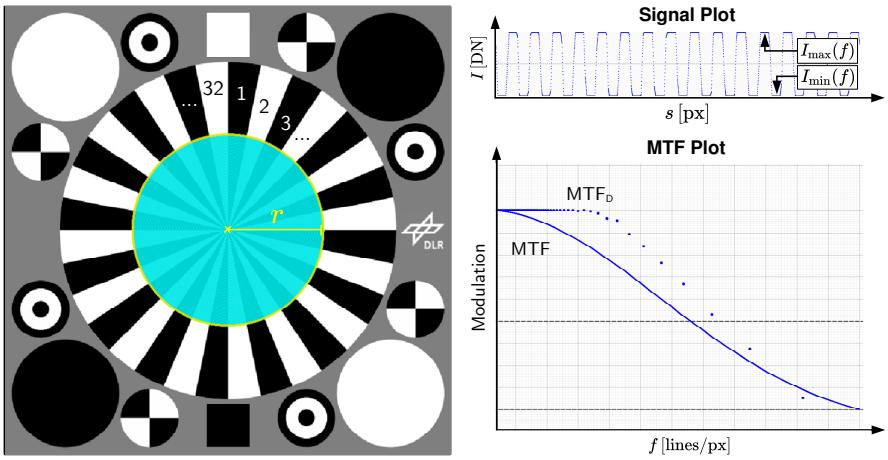


Figure 3.14: *Siemens star Method for two-dimensional blur estimation.* For each circle with radius r that corresponds to a black-white pattern frequency f, pixel intensity contrasts are calculated from $I_{\max}$ and $I_{\min}$ (example top right for yellow radius circle). Post-processing these contrasts forms a discrete MTF_D and fitting a Gaussian function into the discrete values leads to a continuous MTF.

Its continuous version $\mathrm{CTF}\colon [0,1] \to [0,1]$ can be created from by fitting $\mathrm{CTF}_D(f)$ with a Gaussian function $\mathcal{N}(\mu^\star, \sigma^\star)$ using

$$\mu^\star, \sigma^\star \doteq \arg\min_{\mu,\sigma} \int_0^1 \| \mathcal{N}(\mu,\sigma)(f) - \mathrm{CTF}_D(f) \|_2 \, df. \tag{3.24}$$

According to [Col54] and following [Mei20], the CTF forms the camera system response to a square wave input and the MTF the response to a sine wave input. The conversion from CTF to MTF is proposed by a normalization with $\frac{\pi}{4}$ and a series expansion with odd multiples of frequencies f:

$$\mathrm{MTF}(f) = \frac{\pi}{4}\left[\mathrm{CTF}(f) + \frac{\mathrm{CTF}(3f)}{3} + \frac{\mathrm{CTF}(5f)}{5} + \dots\right]. \tag{3.25}$$

3.3.2 Noise Estimation

An image noise process u (3.12) follows $u \sim \mathcal{P}(\sigma^2_{\mathrm{PN}})$ in the case of PN and $u \sim \mathcal{N}(0, \sigma^2_{i\in\{\mathrm{DCSN},\,\mathrm{RN}\}})$ in the cases of DCSN or RN (cf. Sec. 3.2.2). That is, u is determined by the respective standard deviation σ, alias the noise level. Hence, we utilize σ to quantify image noise objectively. The noise levels of the underlying image noise processes of the noise sources can be empirically determined in combination. Due to simplicity, we omit the determination of PN, since the quantum nature of light already decides σ_{PN} given the captured image intensity I (cf. Sec. 3.2.2.1). Thus, we focus on DCSN and RN only. The empirical estimations of $\sigma_{i\in\{\mathrm{DCSN},\,\mathrm{RN}\}}$ base on capturing bias frames and dark frames [Woo15, pp. 163–164]. Let us explain the procedure on Fig. 3.15.

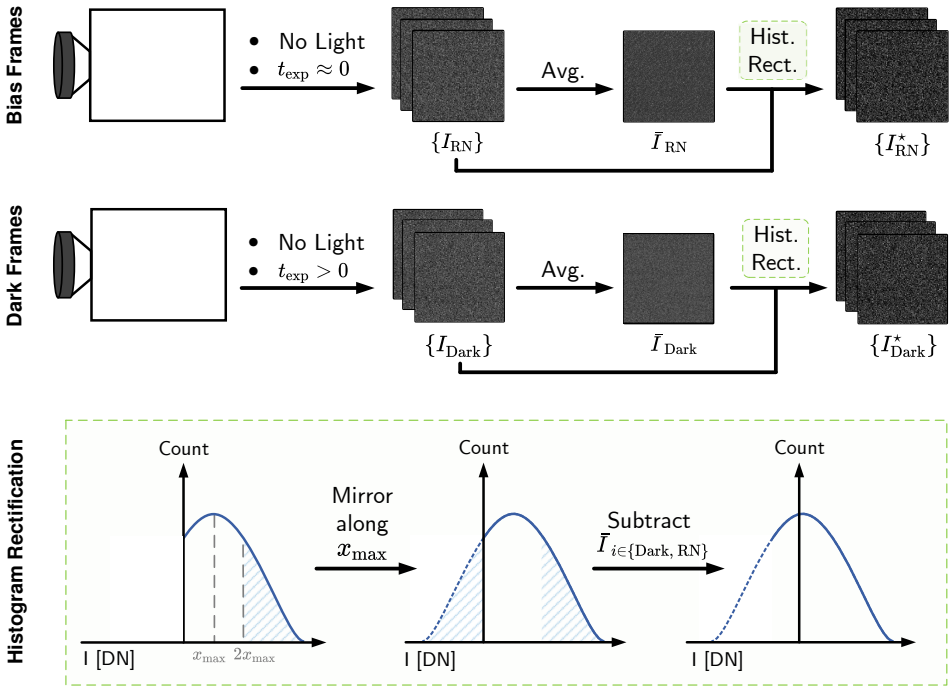


Figure 3.15: *Bias/ dark frame acquisition and post-processing.* Bias frames $\{I_{\text{RN}}\}$ and dark frames $\{I_{\text{Dark}}\}$ are both captured with closed camera shutter, where the exposure time t_{exp} controls dark current accumulation. Averaging the bias/ dark frames results in a so called master bias/ dark frame $\bar{I}_{i\in\{\text{DCSN, RN}\}}$. Bias/ dark frames may be truncated (too low camera offset) and still contain residual additive noise (e.g., dark current), which would both affect noise level estimation. In order to obtain corrected images $\{I^{\star}_{i\in\{\text{DCSN, RN}\}}\}$, we apply a histogram rectification that mirrors values along the distribution maximum x_{max} and subtracts the respective master frame.

A bias frame I_{RN} contains signal bias produced from the camera's readout procedure, i.e., RN. In order to eliminate other signal influences, a bias frame is captured with a closed camera shutter (to prevent signal) and with $t_{\text{exp}} \approx 0\,\text{s}$ (to avoid dark current accumulation). At this point, we identified two common issues that need to be addressed in post-processing: First, the intensity distribution of I_{RN} may be truncated with all negative intensity values set to zero, which results from a small camera-offset and affects the noise level estimation. We tackle this issue by determining the histogram bin x_{max} that corresponds to the distribution maximum and mirroring histogram bins $x \geq 2x_{\text{max}}$ along the vertical axis at x_{max} to reconstruct bins $x \leq 0$. Second, there may be residual fixed-pattern noise in I_{RN} that we rectify as

$$I^{\star}_{\text{RN}} \doteq I_{\text{RN}} - \bar{I}_{\text{RN}}, \tag{3.26}$$

with the rectified bias frame $I^{\star}_{\text{RN}}$ and the mean bias frame $\bar{I}_{\text{RN}}$ (alias master bias frame). Details about the post-processing are specified in Alg. 1 in App. B.2. The underlying RN noise level can be estimated by the sample standard deviation

$$\hat{\sigma}_{\text{RN}} = \sqrt{\frac{\sum_{i=0}^{n}(x_i - \hat{\mu}_{I^{\star}_{\text{RN}}})^2}{n-1}} \qquad \text{with} \qquad \hat{\mu}_{I^{\star}_{\text{RN}}} = \frac{\sum_{i=0}^{n} x_i}{n} \tag{3.27}$$

from a rectified bias frame $I^{\star}_{\mathrm{RN}}$ with n pixel intensities x_i. Averaging the calculated noise levels from multiple (N) bias frames $\{I^{\star}_{\mathrm{RN}}\}_{i=0}^{N}$ increases the robustness of the estimation.

A dark frame I_{Dark} contains any signal bias (assuming radiometrically calibrated cameras: DCSN and RN) and is captured with closed camera shutter and $t_{\mathrm{exp}} > 0\,\mathrm{s}$ to achieve dark current accumulation. The bias post-processing from Alg. 1 in App. B.2 can analogously be applied to I_{Dark}. To obtain the isolated DCSN noise level, we first estimate $\hat{\sigma}_{\mathrm{Dark}}$ from I_{Dark} (cf. (3.27)) and subsequently remove the RN:

$$\hat{\sigma}_{\mathrm{DCSN}} = \sqrt{\hat{\sigma}^2_{\mathrm{Dark}} - \hat{\sigma}^2_{\mathrm{RN}}}, \tag{3.28}$$

following the central limit theorem for the addition of two statistically independent Gaussian distributed random variables [Dev11, pp. 230–232].

3.4 Discussion

Let us briefly discuss our model selections and corresponding limitations of the camera system (Sec. 3.4.1), image quality (Sec. 3.4.2), and image quality assessment (Sec. 3.4.3).

3.4.1 Camera System

The Sec. 3.1 provides a brief overview of the camera system assumed in this work and models to abstract its working principle. Naturally, this section cannot cover the vast variety of camera systems available and each simplification comes with limitations.

Sensor System

The sensor system is the most complex component in a camera system. In addition to the sensor improvements we addressed in the main section, other techniques have been developed to counteract most of the specific drawbacks.

First, the charge transfer of a CCD is not fast enough to avoid electron generation by continuing illumination in the meantime. For this reason, there are also vertical interline or full-frame shift registers, and all shift registers are light shielded [Jan01, p. 8]. Second, anti-blooming drains, different clocking strategies, and interpolation techniques can counteract the addressed blooming, smear, and pixel/row defects [Jan01, ch. 4]. Third, modern large CCDs may have multiple area-wise readout units to speed-up the charge transfer step [Jan01, p. 32]. Fourth, a CIS can comprise readout components (e.g., amplifiers) per row/ column, per pixel, or in a hierarchical setup (e.g., row-s/ column-wise amplification and a second single output amplifier with gain control applied to all pixels) [Wal13]. Lastly, additional circuity for a technique called correlated double (or multi) sampling can reduce noise from the readout circuity to a sub-electron level [Jan01, ch. 6.4].

Please refer to [Jan01] and [HL11] for more information on the mentioned techniques, and CCD and CIS sensor systems in general.

Lens System

The chosen thin lens model suits the majority of camera systems within this study, as lenses are their predominant optical components. However, this model rests upon the geometry of light interpreted as rays and does not take effects from wave-, electromagnetic-, and quantum-optics into account (e.g., diffraction or dispersion). Further, the influence of lens thickness and lens aberrations are excluded. Consequently, many effects known to have a significant impact on the optical processes are neglected and likewise various advances in camera construction to tackle these effects (e.g., lens assemblies or multiple apertures). In addition, this section neither considers special-purpose lens systems (e.g., fisheye-, tele- or zoom-lenses) nor lenses transparent for wavelengths beyond the visible spectrum (e.g., infrared spectrum). For further literature to the aforementioned topics, the reader is referred to [Sun16] and [Hec17].

3.4.2 Image Quality

In the following, we briefly discuss the model selection decisions made to describe images and image quality.

Images

In terms of the sensor processing pipeline from Fig. 3.2, images appear to be a high-level representation of the data produced by the camera system. Having the paradigm of Sensor AI in mind, it might seem counter-intuitive to employ processed data and approve a loss of information content, instead of using the raw voltages directly from the sense node or the amplifier, respectively. Furthermore, the raw voltages are more sensitive to smaller influences of the data quality, which makes it easier to identify, e.g., certain noise sources. This could be important to estimate, for instance, the camera's dynamic range that is typically determined on an electron level [Jan01, p. 113]. On the downside, such low-level data are difficult to interpret and cannot resort to the matured computer vision methods researched in the past decades.

Note that this approach of monitoring the condition in terms of image quality is not limited to monocular camera systems only. This approach also enables the monitoring of multi-view camera systems, whereas the produced image, image quality attributes, and their relation to the camera system might be different (e.g., consider a depth map calculated from stereo images whose intensity values depend on the geometric stereo calibration that is sensitive to concussions).

Image Quality

A major limitation of this work is the choice of potentially too simplified models. For instance: most lens systems depend on temperature. Material stress might change the focal length at runtime and thus change the distortion, which is assumed to be calibrated offline. Another uncovered corner case could be high dark current that saturates the pixels and therefore cannot be mitigated by a calibration anymore. Beside the model choice, there is also a trade-off in the selected metrics to quantify image quality attributes. High-level metrics that incorporate more than one attribute (e.g., signal-to-noise ratio or dynamic range) are more represented in the literature and may be better suited for comparison to other studies. However, with the condition monitoring goal in mind, we decided to employ single-attribute metrics in order to provide a clear link to a respective root cause. The interested reader is referred to [PE18, ch. 2, ch. 3] for further readings about image quality.

Blur

The defocus blur kernel from (3.7) assumes a locally negligible influence of scene depth (as we aim for an image-patch-wise image quality estimation) and hence a uniform distributed circular blur kernel. In the case of arbitrary environments, this simplification turns out to be suitable if the related parameters of (3.6) minimize the scene depth influence (i.e., by focusing the hyperfocal distance), or if a fixed image-patch-size approaches the coverage of a small part of the scene with low depth diversity (i.e., by a larger focal length). In case of a significant scene depth influence, a Gaussian-shaped defocus blur kernel is typically used for approximation. However, all defocus blur models working on the image plane have known limitations in comparison to a superior 3D scene description [Dem04]. Other defocus and motion blur influences outside of our scope are, for instance, a low mechanical shutter speed, a rolling shutter, or pixel crosstalk [PE18, p. 17, pp. 42–43, p. 88].

Noise

The proposed noise types are known to have the greatest impact in most applications, but represent only a small fraction of the various possible types. Please consider [Jan01, ch. 7, ch. 8] and [HL11, ch. 6] for details.

3.4.3 Image Quality Assessment

When using the proposed noise and blur estimation methods, it is necessary to pay attention to the following details that limit their application.

Blur Estimation

The results of both the slanted-edge method and the Siemens star method depend on not yet standardized aspects including: size of the employed image edge(s), the edge's

position on the image plane, rotation of the target edge outside of the image plane, and image quality effects that influence image intensities (noise, color fringing, etc.). These varying conditions limit the comparability among different blur estimation studies. Meißner recently researched such standardization issues and proposed approaches that will be included in the upcoming German standard DIN 18740-8 [Mei20; BR17]. However, both methods propose the usage of standardized target patterns (e.g., the Siemens star pattern) in order to fulfill the requirements on a target edge to analyze, which is not practical in field operations.

Noise Estimation

A proper noise estimation requires a radiometrically calibrated camera system, since uncalibrated residual effects can limit the estimation result. Please consider that the usage of master bias/ dark frames to tackle residual effects can only counteract additive residuals (e.g., dark current) but not multiplicative ones (e.g., pixel non-uniformities) [Woo15, pp. 205]. To this end, on-board calibration equipment have been proposed for automatic radiometric calibration during field operation [CS96]. An alternative approach to estimate RN/ DCSN is to read-out overscan pixels (sensor pixels that are not illuminated)[Jan07, pp. 50–51], however, most camera systems do not provide access.

3.5 Summary

In this chapter, we introduced physical models and standard blur/ noise estimation approaches that we employ within this thesis.

In Sec. 3.1, we assumed a simple camera model that consists of two main components: a sensor system and a lens system. For the sensor system, we outlined its general functionality to sense and digitize incident light, and subsequently provided a brief overview of the two predominant sensor types (CCD and CIS), including respective advantages and resulting fields of application. Analogously, we introduced the working principle of a lens system to focus incident light onto the sensor on the example of the assumed thin lens model.

In Sec. 3.2, we addressed image quality as a mean to approach a camera's condition on the basis of its produced image data. From the variety of existing image quality attributes, we put focus on blur and noise as two of the most common ones researched in the literature. Thereupon, we detailed their root causes on a physical basis and respective parameters to control the effects. Specifically, we emphasized defocus blur and motion blur, and photon shot noise, dark current shot noise and readout noise as the most important root causes relevant to autonomous machines.

Section 3.3 introduced the concept of predictive image quality assessment to quantify individual image quality attributes (blur and noise) in order to estimate image quality. To this end, we presented two objective assessment metrics: the modulation transfer function (MTF) for blur and the noise level (σ) for noise. Moreover, we demonstrated the slanted-edge method and the Siemens star method as two standard means to estimate an MTF, and a noise estimation procedure that relies on rectified bias and dark frames.

Finally, we end this chapter with a brief discussion to point out respective shortcomings of the chosen models and estimation methods, and to provide outlooks beyond our thesis scope with further literature (Sec. 3.4).

CHAPTER 4

Camera Self-Health-Maintenance Framework

This chapter introduces our framework to self-maintain the intended functionality of a camera system. We first briefly recapitulate the objectives and the intended scope for the framework (Sec. 4.1). Subsequently, we summarize the framework on a high level of abstraction (Sec. 4.2) and detail its two main components: a condition estimation (Sec. 4.3) and a decision & control unit (Sec. 4.4). For the condition estimation, we first present employed methods to quantify total image blur and noise for the case of total blur/ noise assessments. On this basis, we propose a novel approach to identify and quantify the contributions of multiple noise sources. For the decision & control unit, we suggest means to decide on countermeasures, which we can achieve by controlling camera parameters. Lastly, we discuss on the proposed framework (Sec. 4.5) and summarize this chapter (Sec. 4.6). This chapter is partially published in [Wis+23b] and [Wis+23a].

4.1 Requirements

Let us first repeat the objectives and the scope for the framework design (cf. Sec. 1.1):

- **Autonomous Mobile Machines** operate in different environments with limited computational resources. Thus, they require a general and reliable framework that can run on mobile hardware in real-time. Our focus lies on panchromatic cameras that operate in the visible range.
- **Self-Health-Maintenance** implies the automatic estimation of the current condition and, in case of a detected misbehavior, the automatic initiation of countermeasures.
  - **Condition Estimation** can be approached by assessing the quality of produced image data as a proxy. Image quality is determined by multiple image attributes (blur and noise), the weighting of which depends on the envisaged high-level image application (object detection). Optimizing high-level application performance requires objective image quality measures.

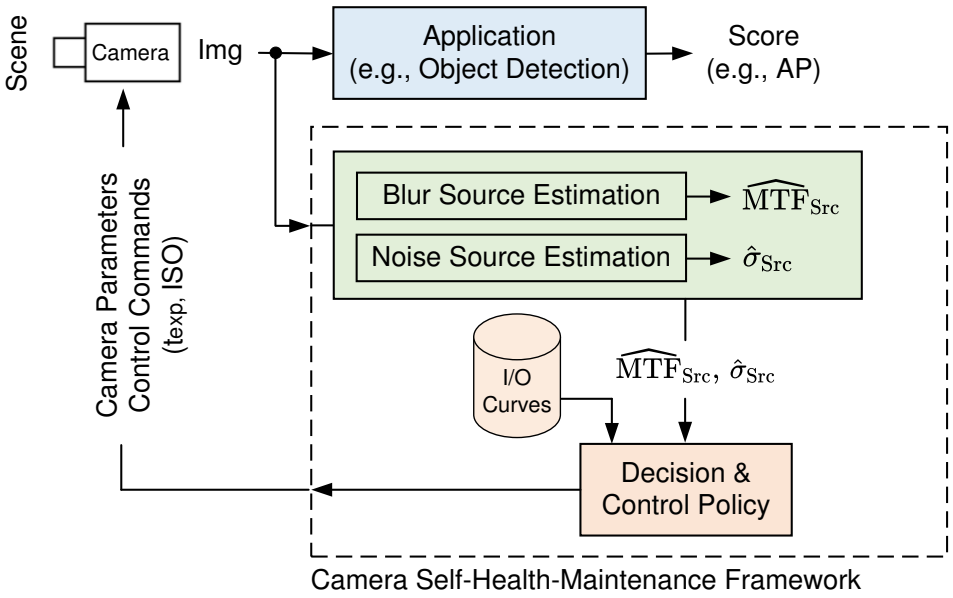


Figure 4.1: *Camera self-health-maintenance framework: overview.* The camera is constantly monitored by analyzing image corruptions (e.g., blur and noise). According to the estimated severity and the underlying root causes of corruptions, camera control parameters (e.g., exposure time t_{exp} and ISO gain) are recalculated to maximize application performance using the (offline determined) input/output (I/O) performance curves.

In order to initiate suitable countermeasures for a detected misbehavior, problem root causes need to be identified. We focus on time-varying and region-wise root causes that originate during image digitization inside the camera system.

– **Countermeasures** aim to optimize target application performance. To this end, the framework needs knowledge about the relation between image attributes and target application performance, and access to the camera to control image attributes (blur and noise). This knowledge must be available at runtime, so we obtain it offline. We focus on motion blur in the presence of noise.

- **Sensor AI** "incorporate(s) sensor knowledge which is gained by modelling, characterization and application into data analysis" in form of a "holistic approach which considers entire signal chains from the origin to a data product" [Bör+20].

4.2 Overview

We propose a camera self-health-maintenance system that consists of online testing (Fig. 4.1) and offline training parts (Fig. 4.2).

Let us introduce the offline training procedure first. We start with image datasets from a target application domain as input (e.g., object detection) and corrupt them according to an image formation pipeline (Fig. 3.7 on p. 34). The pipeline contains the

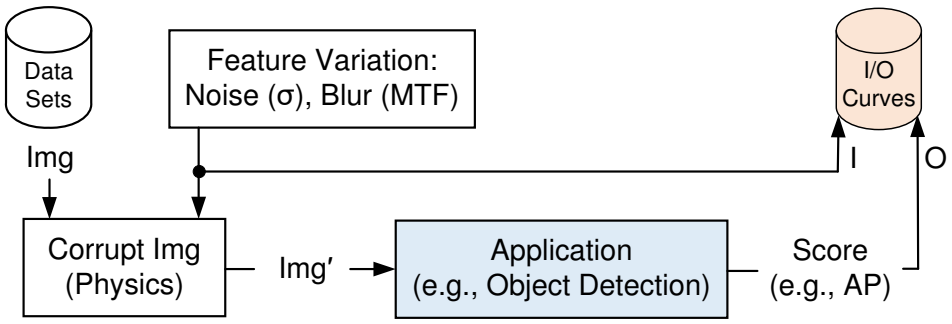


Figure 4.2: *Camera self-health-maintenance framework: training.* An offline sensitivity analysis determines the impact of physical image corruptions (e.g., blur and noise) on the performance of a target application (e.g., object detection), and stores the results in input/output (I/O) performance curves. As input, image data close to the application domain are used.

most common (physics-based) sources of blur and noise affecting the camera condition, with realistic severity levels (see Sec. 5.1.1). We quantify these levels using noise level σ and modulation transfer function (MTF) values, respectively. Afterwards, we let our system's target application (object detection) evaluate these corrupted images. We likewise quantify this performance in terms of the well known objective average precision score [Eve+09] (AP, Sec. A.1). Knowing each applied image corruption and the corresponding calculated application performance, the respective tuples are aggregated into input/output performance curves (IOPC), which is the final product of this training procedure.

The testing part (Fig. 4.1) has access to these IOPCs and analyzes each captured (yet unprocessed) camera image online. We distinguish between the cases of isolated and multiple blur/ noise sources. For the first case, we adapt existing machine-learning–based, real-time capable noise level and MTF estimators (Secs. 4.3.1 and 4.3.2), and evaluate their estimation and runtime performances compared to traditional state-of-the-art estimators for combined and individual corruption cases (Secs. 5.2 – 5.4). Thereupon, we introduce a simple approach to repair blur estimation in case of interfering high noise levels (Sec. 5.5). For the second case, we propose a novel approach to identify and quantify multiple noise sources on the basis of an image and corresponding camera metadata (Sec. 4.3.3). If the estimated image quality does not meet the requirements for optimal application performance recorded in an IOPC, a control policy decides how to adjust camera parameters as countermeasure. We propose two exemplary control policies using exposure time and ISO gain to trade off blur and noise. They exploit the fact that object detectors are typically more sensitive to blur than to noise (Sec. 7.1).

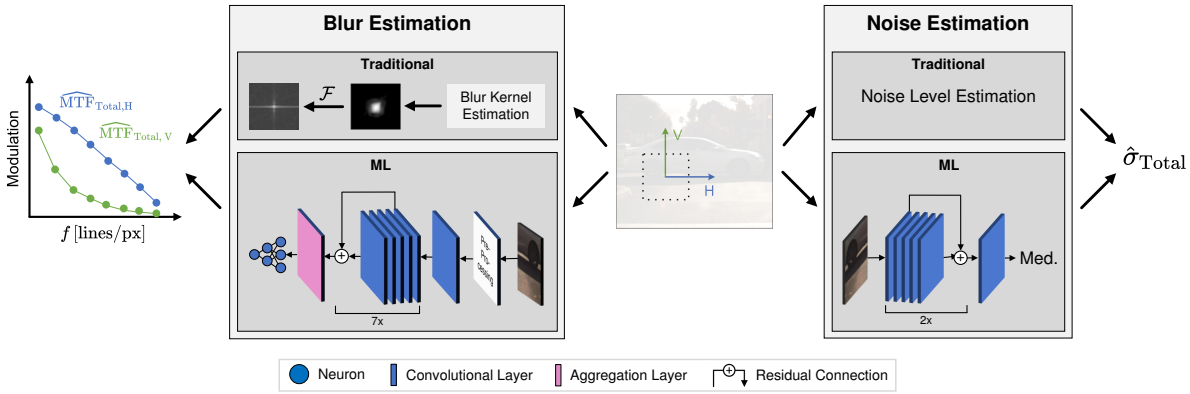


Figure 4.3: *Total blur and noise estimators.* Traditional (top branches) and learning-based (bottom branches, ML) total blur/ noise estimation approaches. Blur estimation: All methods take one or more image patches as input and output estimated MTF samples for pre-defined image *frequencies* (f) in the *horizontal* (H) and *vertical* (V) directions. Traditional methods first estimate a blur kernel, transform it into Fourier space $\mathcal{F}$ and sample MTF values. The learning-based method consists of a *pre-processing* stage followed by a multi-layer CNN. Noise Estimation: Both approaches estimate a total noise level $\hat{\sigma}_{\text{Total}}$ for each input image patch. The learning-based method employs a CNN whose output is the median over pixel-wise estimations to obtain a single noise estimate.

4.3 Condition Estimation

We start with a presentation of employed blur and noise estimators to quantify the respective image attributes objectively without root cause identification (Secs. 4.3.1 and 4.3.2, Fig. 4.3). We summarize both traditional and learning-based (ML) approaches first and then detail our improvements. Subsequently, we propose a novel approach to identify and quantify the contributions of individual noise sources (Sec. 4.3.3). For the sake of completeness, we address the problem of blur source estimation in a discussion at the end (Sec. 4.5.1).

4.3.1 Blur Estimation

The goal of our image blur estimators is to predict the total MTF given a possibly blurred input image patch I^*, which is assumed to be monochrome (i.e., grayscale) and of size 192×192 pixels (following the ML approach). The left part of Fig. 4.3 summarizes the steps of the two main paradigms.

Traditional methods (non-learning–based)

We use two baseline methods: "*graph-based*" [Bai+18] (GBB) and "*simple local minimal intensity prior*" [Wen+20] (PMP) as traditional blur kernel estimators (top left branch in Fig. 4.3). Both estimators follow a maximum-a-posteriori framework

$$\min_{I,h} \mathcal{L}(I \circledast h, I^*) + \alpha G(I) + \beta R(h) \tag{4.1}$$

to iteratively refine a clean latent image I and the blur kernel h (3.5). The objective function (4.1) is the negative logarithm of the posterior distribution (thus maximization turns into minimization). It consists of a data fidelity term ($\mathcal{L}\colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$) that penalizes the deviations with respect to the observed image I^*, and two regularizers $G\colon \mathbb{R}^2 \to \mathbb{R}$ and $R\colon \mathbb{R}^2 \to \mathbb{R}$ (prior knowledge) on the unknowns (with positive weights α, β). The GBB method [Bai+18] represents images as graphs and employs a skeleton image with only strong gradients as a proxy for I. It uses a re-weighted graph total variation prior $G(I)$ to favor bi-modal image histograms. The PMP method [Wen+20] builds on top of the dark-channel prior, proposing a simplified patch-wise minimal pixel prior $G(I)$ that aims for sparse minimal pixel intensities with small computation complexity. The resulting h from each method is Fourier-transformed into the MTF and sampled at the same spatial frequencies as the learning-based approach, for better comparison. We use the source code from [BYc18; FWe19], setting the kernel size parameter to 31×31 pixels in order to estimate large blur kernel.

Learning-based Method

We upgrade a learning-based approach [Bau+18] to *directly* estimate MTF values of camera lenses from natural images (without estimating the kernel h first). It consists of a pre-processing stage followed by a CNN.

The pre-processing stage includes four steps: (*i*) Intensities are first scaled to $[0, 1]$ and mean-normalized. (*ii*) A rotation is applied for estimations of the MTF in radial and tangential directions. (*iii*) The Sobel-filtered image patch is passed as an additional channel to aid the MTF estimation procedure. (*iv*) Channels are spatially downsampled to enlarge the receptive field of early CNN layers. We alter step (*ii*) to distinguish between estimations in horizontal and vertical directions[1], and thus be able to compare to baseline methods GBB and PMP.

The CNN consists of a convolutional layer, seven residual blocks with strided convolutions, an intermediate feature representation layer that allows an averaged feature activation of optional multiple input patches, and three fully connected layers that regress the MTF outputs (bottom left branch of Fig. 4.3). The resulting output consists of eight MTF values at pre-defined spatial image frequencies $f \in \{0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5, 0.6\}$ lines/px.

The training is supervised. In [Bau+18], pairs of sharp image patches and PSFs (I, h), synthetic or real, are collected. Their convolution leads to the training samples I^*; and the respective MTF samples of the PSFs at pre-defined frequencies are the training labels. In contrast to [Bau+18], we blurred the sharp images by simulated random defocus and motion blur kernels (kernel models from Sec. 3.2.1 and parameters from

[1]This method can estimate MTFs in any direction. For simplicity, we focus only on two directions.

Sec. 5.1.1) and retrained the CNN with default parameters. In this way, the CNN is expected to learn to specialize in defocus and motion blur estimation.

At inference time, we pass a batch of four input image patches, i.e., we stack temporally consecutive patches from the same sensor position, pre-process them independently and input them into the CNN at once. We expect better results this way according to [Bau+18], although one patch works as well. The obtained CNN output is then an (averaged) MTF estimation.

Since the original source code is not available, we re-implemented it with guidance from the authors, who also provided their original training data. The final network comprises 6.76M parameters.

4.3.2 Noise Estimation

The goal of the image noise estimators is to predict the noise level σ of a noise process given a noisy input image patch $\tilde{I}$, which is monochrome and of size of 128×128 pixels (following the ML approach). The right part of Fig. 4.3 depicts the steps of the two main paradigms.

Traditional methods (non-learning–based)

As baseline estimators we use the works of [Shi+05] (self-implemented) and [CZA15] (with its code basis [Yue19]). Both are representatives of the two major noise estimation approaches in the literature:

The *adaptive Gaussian filtering method* [Shi+05] (B+F) uses the standard deviation of the most homogeneous image patches as a basis to calculate a Gaussian kernel that is used to filter such patches. The standard deviation of the difference between filtered and unfiltered patches leads to the estimated $\hat{\sigma}_{\text{Total}}$. We increased the internal image patch size from 3×16 to 8×16 pixels due to better observed results on the selected datasets.

The method [Yue19] decomposes image patches via *Principal Component Analysis* (PCA, also abbreviation of the method) into their eigenvalues and assigns the noise ratio to the smallest ones. In contrast to previous work, the authors tackle the problem of overestimating or underestimating noise theoretically and propose an efficient non-parametric algorithm for noise level estimation.

Learning-based Method

We use the deep residual noise-level estimator (DRNE) of [Tan+19; Tan18] as learning-based (ML) approach. It was designed for pixel-wise noise level estimation from signal-dependent noisy images. The noise model was Gaussian with parameters that accounted for photon and readout noise.

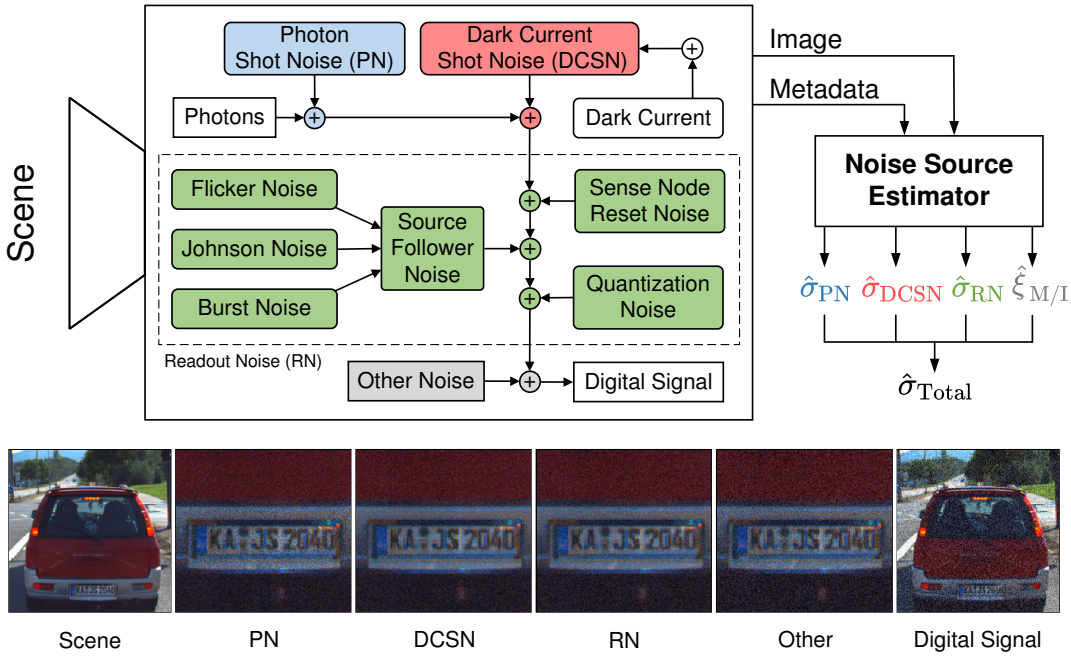


Figure 4.4: *Proposed noise source estimation.* Different noise sources affect the image formation process of a scene. Our noise source estimator quantifies major noise source contributions $\hat{\sigma}_{i\in\{\mathrm{PN,\ DCSN,\ RN}\}}$, unexpected noise $\hat{\xi}_{\mathrm{M/I}}$, and the total image noise $\hat{\sigma}_{\mathrm{Total}}$ using an image and camera metadata.

The CNN consists of 16 convolutional layers (including three residual blocks) and lacks pooling and interpolation layers due to a known performance decrease for image noise tasks. The resulting output $\hat{\sigma}_{\mathrm{Total}}$ is estimated for each pixel, but for a better comparison with baseline methods we use the average over the patch as the noise level estimator. The CNN contains a total of 519k parameters.

Training in [Tan+19] is supervised, carried out by artificially adding noise with $\sigma_{\mathrm{Total}} \in [0, 30]$ to images from the Waterloo dataset [Ma+16]. We retrained the CNN in the same way using our noise model from Sec. 3.2.2 and parameters specified in Sec. 5.1.1.

4.3.3 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

$$\sigma_{\mathrm{Total}} = \sqrt{\sigma^2_{\mathrm{PN}} + \sigma^2_{\mathrm{DCSN}} + \sigma^2_{\mathrm{RN}}} + \xi_{\mathrm{M/I}} \tag{4.2}$$

and its individual components: the photon shot noise (PN) level σ_{PN}, the dark current shot noise (DCSN) level σ_{DCSN}, the readout noise (RN) level σ_{RN} and a component $\xi_{\mathrm{M/I}}$ that quantifies unexpected (i.e., residual) noise (Fig. 4.4). We assume grayscale patches, of size 128×128 px. Next, we describe the base architecture, subsequently detail our extensions, and lastly focus on training the noise source estimator.

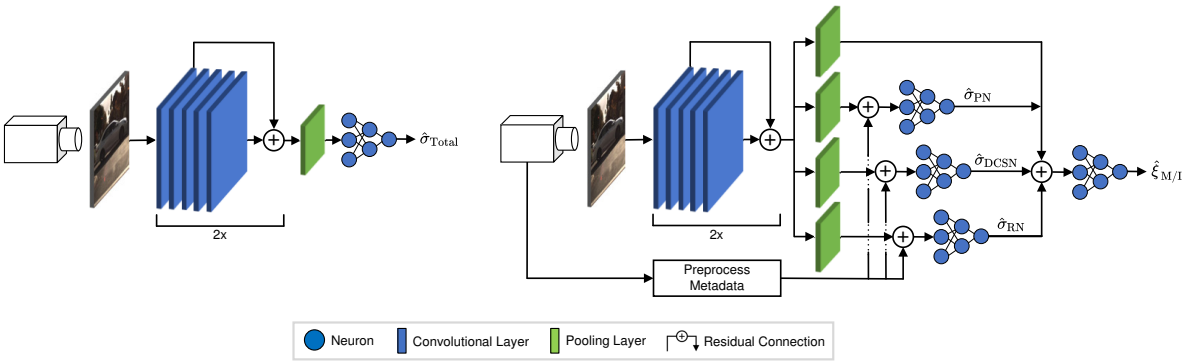


Figure 4.5: *Noise level estimator vs. proposed noise source and level estimator.* Left: Customized baseline estimator $DRNE_{\text{cust.}}$, which predicts the noise level of the input image's total noise. Right: Proposed noise source estimator that additionally employs camera metadata and predicts the noise levels of four different noise types.

Base Architecture

Our method is inspired by the DRNE from [Tan+19] (Sec 4.3.2). We customize its architecture so that the neural network takes grayscale images as input and estimates only one noise level per image patch (left part of Fig. 4.5). Specifically, we replace the first $3 \times 3 \times 3$ convolution kernel by a 3×3 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 519k to 336k while achieving similar estimation accuracy as DRNE. Lastly, we retrain the network as described in Sec. 4.3.3. In the upcoming sections we refer to this customized model as $DRNE_{\text{cust.}}$.

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. In the following, we describe the three major extensions to the above method for noise *source* estimation (right part of Fig. 4.5): noise type identification (with or without the inclusion of camera metadata), and quantification of unexpected noise.

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.

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 Tab. 4.1). We assume the fixed metadata to be constant at training and inference times due to multiple reasons: (*i*) Only parameters that in a sensitivity analysis lead to significant noise changes in the noise model are picked as variable parameters (see App. B.2).

Variable Parameter	Value Range	Variable Parameter	Value Range
Minimal Metadata		**Full Metadata (cont.)**	
Camera Gain	[0, 24] dB	Pixel Clock Rate	[8, 150] $\times 10^{6}$ Hz
Exposure Time	[0.001, 0.2] s	Sense Node Gain	[1, 5] $\times 10^{-6}$ mm
Sensor Temperature	[0, 80] °C	Sense Node Reset Factor	[0, 1]
		Sensor Pixel Size	[0.0009, 0.01] mm
Full Metadata		Sensor Type	{CCD, CMOS}
Dark Signal FoM	[0, 1]	Thermal White Noise	[1, 60] $\times 10^{-9}$ Hz
Full Well Capacity	[2, 100] $\times 10^{3}$ e^{-}		

Table 4.1: *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 App. B.2.

From these parameters, we also fix (*ii*) 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[2] 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 the range $[0, 1]$ (using their value ranges in Tab. 4.1). 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: (*i*) fast (using a GPU), (*ii*) allows us to train on real noise data that is not covered by the noise model, and (*iii*) allows us to perform non-trivial feature-wise fusion with the feature maps from the processed input image.

3. Unexpected Noise Quantification. In the proposed system (right part of Fig. 4.5), 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: (*i*) corrupted metadata (e.g., by camera malfunctioning) or (*ii*) unmodeled noise sources (e.g., also by hardware damages, or a general mismatch between the noise model and the real image noise).

[2]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.

Specifically, we train this fourth FCB to quantify

$$\begin{aligned}\xi_{\mathrm{M/I}} &\doteq \sigma_{\mathrm{Model}} - \sigma_{\mathrm{Image}} \\ &\overset{(4.2)}{=} \sqrt{\sigma^2_{\mathrm{PN}}(M_1) + \sigma^2_{\mathrm{DCSN}}(M_2) + \sigma^2_{\mathrm{RN}}(M_3)} - \sqrt{\sigma^2_{\mathrm{PN}}(M'_1) + \sigma^2_{\mathrm{DCSN}}(M'_2) + \sigma^2_{\mathrm{RN}}(M'_3)}\end{aligned} \tag{4.3}$$

with $\xi_{\mathrm{M/I}}$ normalized to $[-1, 1]$ for training, the total image noise σ_{Image}, the total modeled noise σ_{Model}, and metadata sets $M_1, \ldots, M_3$, and altered sets $M'_1, \ldots, M'_3$ having a different randomly generated camera gain. The metadata sets $M_{i\in\{1,2,3\}}$ are only fed to the FCBs (corresponding to noise level σ_{Model}) while the altered sets $M'_{i\in\{1,2,3\}}$ are used to corrupt the image (with corresponding noise level σ_{Image}). In this way, the network learns to capture the mismatch between the metadata and the image noise in $\xi_{\mathrm{M/I}}$.

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

Training Details

We utilize an almost noise-free dataset with natural images (TAMPERE21 [BPE22]), 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 afterwards corrupted with noise generated by the noise model of [KW14]. Each image patch is corrupted independently with its own set of randomly generated variable metadata. In this way we generate ≈103k 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. Further implementation details and the training configuration can be found in App. B.2.

4.4 Decision and Control Policy

This section introduces an approach to determine the dependencies of a target application (object detection) on image quality attributes (blur and noise) (Sec. 4.4.1). On this basis, we detail a camera parameter adjustment routine to trade off blur and noise in order to maximize application performance (Sec. 4.4.2).

4.4.1 Object Detection Sensitivity Analysis

We choose object detection as a representative modern image application of great importance in various fields. And, as presented, we choose image noise and blur as the main data quality indicators of the state of our imaging system.

Specifically, we use YOLOv4-416 [BWL20] and Faster R-CNN [Ren+15] as state-of-the-art real-time object detectors (with pre-trained models and default settings, applied on grayscale images).

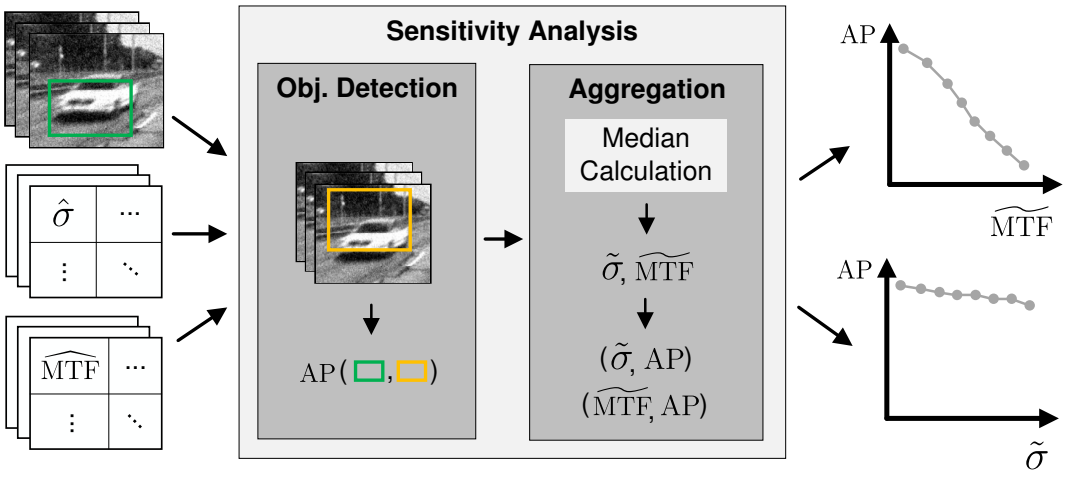


Figure 4.6: *Sensitivity analysis of object detector performances for blur and noise.* The detectors are evaluated on corrupted images resulting in average precision (AP) scores. For the true detection areas, corresponding patch-wise noise ($\hat{\sigma}$) and blur ($\widehat{\text{MTF}}$) estimations are aggregated to medians ($\tilde{\sigma}$ and $\widetilde{\text{MTF}}$) and, together with the APs, added to performance curves.

The goal is to determine object detection sensitivities *empirically* for different noise and blur levels of considered root causes (Fig. 4.6). For the sake of simplicity, we do not consider specific blur or noise sources in this section. For a condition monitoring application, we require this performance curves to be available at runtime, so we perform this analysis offline. It is also feasible to build or refine the performance curves iteratively in an online approach, however, at the expense of controlled image blur/ noise.

Let us explain the offline procedure on the example of a fixed image blur level MTF and noise level σ. As input we assume N_I images $\{I\}_{i=1}^{N_I}$ with N_{GT} ground truth object detections $\{B_{\text{GT}}\}_{i=1}^{N_{\text{GT}}}$, and corresponding patch-wise blur $(\cup_{jk}\{\widehat{\text{MTF}}_{jk}\}_{i=1}^{N_I})$ and noise estimations $(\cup_{lm}\{\hat{\sigma}_{lm}\}_{i=1}^{N_I})$ estimations, where j, k, l and m address respective patches (Fig. 4.6). First, both object detectors are applied on the images to gather estimated detections $\{B_{\text{D}}\}_{i=1}^{N_{\text{D}}}$. Second, these estimations are scored with the well-known average precision (AP) metric [Eve+09]. We provide details to the AP calculations in App. A.1.

In a subsequent aggregation step, we determine the corresponding median noise and blur estimations of all the respective image patches overlapping with the ground truth object detections:

$$\begin{aligned} \widetilde{\text{MTF}} &\doteq \text{median}(\{\widehat{\text{MTF}}_{jk} \,|\, \widehat{\text{MTF}}_{jk} \in \cup_{jk}\{\widehat{\text{MTF}}_{jk}\}_{i=1}^{N_I} \wedge I_{jk} \cap B_{jk,\text{GT}} \neq \emptyset\}) \\ \tilde{\sigma} &\doteq \text{median}(\{\hat{\sigma}_{lm} \,|\, \hat{\sigma}_{lm} \in \cup_{lm}\{\hat{\sigma}_{lm}\}_{i=1}^{N_I} \wedge I_{lm} \cap B_{lm,\text{GT}} \neq \emptyset\}) \end{aligned} \tag{4.4}$$

To bound the complexity, we quantize the estimation parameter spaces into bins with $\tilde{\sigma} \in \{0, 5, \ldots, 30\}$ DN and $\widetilde{\text{MTF}} \in \{0.1, 0.2, \ldots, 1.0\}$.

Finally, the resulting input-output tuples $(\tilde{\sigma}, \text{AP})$, $(\widetilde{\text{MTF}}, \text{AP})$ or $(\tilde{\sigma}, \widetilde{\text{MTF}}, \text{AP})$ are collected as *performance curves* (IOPCs).

4.4.2 Optimizing Object Detection by Trading off Blur and Noise

We now demonstrate how one can use the online blur/ noise estimators and the offline empirical input-output performance curves to control image quality and hence optimize the system's performance (Fig. 4.1). We focus on actions tackling linear motion blur (LinMB) here because object detectors are substantially more sensitive to LinMB than to noise (cf. Sec. 7.1), and there is abundant motion blur in standard datasets like Udacity (Fig. 5.6).

We make the following considerations knowing the camera's physical process. The main controllable influencing factor of motion blur is the camera's exposure time t_{exp} (Sec. 3.2.1.2). We exploit the relations

$$\begin{gathered} t_{\text{exp}} \propto I \text{ and } t_{\text{exp}} \propto \text{MB} \sim \text{MTF}^{-1} \sim \text{AP}, \\ \text{ISO} \propto I \text{ and ISO} \propto \sigma \sim \text{AP}^{-1}, \end{gathered} \tag{4.5}$$

where ISO denotes the camera ISO gain and AP is the average precision of the object detector.

Changing t_{exp} by a factor

$$\alpha \doteq t_{\text{exp}}^{\text{old}} / t_{\text{exp}}^{\text{new}} \tag{4.6}$$

equally changes the aggregated amount of light intensity I (assuming sensor linearity) and also MB by the same factor (assuming constant relative speed between camera and scene). To compensate for the changed light, we may alter the camera ISO gain by factor α, which likewise changes the noise level σ. This relationship depends on the camera sensor architecture and whether the analog or digital signal is amplified [Igu19]. We assume digital amplification as the worst case and thus a linear relation. Consider that $t_{\text{exp}} \propto \sigma_{DCSN}^2$ during optimization as well (cf. Sec. 3.2.2.2).

Hence, we can model the problem as an optimization one, i.e., determining α from the IOPCs to maximize the object detector's score:

$$\alpha^\star = \arg\max_{\alpha} \ \text{AP}(\alpha\,\hat{\sigma}, \alpha\,\text{MB}_L(\widehat{\text{MTF}})). \tag{4.7}$$

Note that the relation $\text{MB} \sim \text{MTF}^{-1}$ may be non-linear, so we optimize the MB size MB_L that corresponds to an estimated MTF.

4.5 Discussion

Let us briefly discuss the framework design based on its individual components: the *condition estimation* (noise, noise sources, and blur estimation) and the *decision & control* units.

4.5.1 Condition Estimation

Here we first address the proposed blur and noise (source) estimators. Subsequently, we discuss on their applicability to general camera systems and consider alternative Sensor AI approaches. Lastly, we reference a method for blur source estimation.

Blur/ Noise Estimation

In most approaches, blur/ noise estimation and deblurring/denoising are inseparable. One the one hand, this limits the application to fight symptoms only and on the other hand dedicated blur/ noise estimation comes with an overhead of computational cost. This holds especially for the used traditional blur estimators that rely on the maximum-a-posteriori framework, which prevents real-time capability and thus limits their comparability to the proposed learning-based approach (cf. Sec. 5.2.1).

On the downside, machine learning methods are still considered black boxes, despite recent advances in analyzing training and inference procedures [TG20]. As a consequence, they can still produce unexpected results, for instance, in the case examples not seen during training (out-of-domain examples), and thus decrease a reliable operation of related mobile machines. However, researchers propose techniques to quantify and decrease uncertainty of learning-based methods [Gaw+21]. To this end, we evaluate a temporal aggregation of estimation results in our experiments (Sec. 5.6).

Noise Source Estimation

Let us consider an alternative approach to fuse the input image and the metadata information: It would also be conceivable to input both at the first layer of the CNN and to omit individual network branches, that is, to neglect the integration of prior knowledge that relates certain metadata to independent noise sources in the network's architecture. In this way, the network could learn more powerful combined image–metadata features and the network size could be reduced. However, as indicated in the work of [Wil+22], prior knowledge about the physical processes helps the network to learn the desired model and helps to counteract its black-box character.

Generality

The proposed noise and noise source estimators are applicable to sensors that follow the noise model [KW14] for which they were trained, i.e., standard CCD and CMOS sensors

introduced in Sec. 3.1.1. Deviating sensor components (e.g., germanium photodiodes [Kau+11]) can lead to different noise statistics (e.g., fundamentally larger dark current noise [Kau+11]). In addition, different noise types can be expected for sensors that do not rely on the photoelectric effect to sense photons (e.g., film grain noise for photographic film [NS78] or thermal fluctuation noise for microbolometers [Ric94, p. 8]). Both limit the applicability of our estimators.

The blur/MTF estimation is not limited to a camera system design, as it operates on the image plane and does not rely on model assumptions. The estimation procedure can also be easily extended to cameras with multiple spectral bands by converting the resulting multi-channel image into a grayscale one or by evaluating each band separately. From an application perspective, MTF estimation relies on scene objects to serve as image features. As a result, applications in scenes with large homogeneous areas (e.g., under- or over-illuminated areas, sky, space, underwater areas) limit the applicability.

Sensor AI

We introduce different approaches to combine data-based and physics-based paradigms (Sensor AI) in Sec. 2.3.2. On this basis, all of our proposed learning-based estimators can be classified as *initialization* combination methods. Moreover, our noise source estimator also belongs to the *architecture* class. There are alternative ways to combine both paradigms as well.

A different approach would be to integrate physical relationships into the loss functions at training time, for example, one could add regularization terms to encode relationships between the blur/ noise levels and the corresponding camera metadata explicitly. On the one hand, previous studies have shown this could help to learn the desired physical model and may reduce needed training data [Wil+22]. On the other hand, it would prevent the possibility to fine tune on real-world data that is not covered by the respective physical model.

Blur Source Estimation

The problem of blur source estimation is not covered in this thesis. We refer the interested reader to the study of Tiwari *et al.* [Tiw+14] for an existing approach that relies on blur pattern analysis in the frequency spectrum of an image. They propose to estimate defocus blur radius, motion blur length, and motion blur angle from images that are corrupted by joint defocus and motion blur. For the first two parameters, they employ a generalized regression neural network that inputs a pre-processed image and for the latter parameter a simple processing of a radon transformed image. However, the authors evaluated their approach only on bar code images with simulated blur.

4.5.2 Decision and Control Policy

In the following, we address the limitations of considering only two image quality properties (blur and noise) and one target application (object detection) in our framework. Analogously to Sec. 4.5.1, we further discuss on the applicability to general camera systems.

Object Detection Sensitivity Analysis

The effectivity of the framework is limited in how careful the training dataset is created. This includes how well the training dataset covers the target application domain, how diverse the data is within the application domain, and how large the simulation-to-reality gap is in the case of simulated image quality attribute effects [Rew+20].

Moreover, target application performance can also depend on further image quality attributes besides blur and noise. The studies [Mic+19; HD19] benchmark exemplary image effects for object detection (such as image intensity changes that we bypass by compensating t_{exp} changes with the ISO gain). Each additional attribute we consider in the sensitivity analysis increases the problem's domain significantly (the necessary data to analyze increases exponentially). A solution to this dimensionality problem could be to investigate conditional independencies of image quality attributes on the estimators and on the target application performance, so that attributes do not need to be analyzed in combination (cf. [Dev11, p. 83]). Moreover, the sensitivity analysis is highly parallelizable – each image data subset that corresponds a fixed image quality attribute value configuration can be computed independently.

Extension to multiple Target Applications

Our proposed decision & control policy assumes that the camera images are used exclusively for object detection, i.e., a single target application. However, mobile machines typically employ camera images for multiple high-level applications to support their actions at runtime (navigation, environment mapping, etc.). Our proposed framework can be generalized to optimize multiple application performances in combination. Therefore, we perform the sensitivity analysis for all desired applications separately (corrupted images can be re-used within the same application domain). The optimization in (4.7) can then be extended to

$$\alpha^\star = \arg\max_{\alpha} \sum_{i=0}^{n} \lambda_i M_i \left(\alpha\, f_i(\hat{\sigma}), \alpha\, g_i(\widehat{\text{MTF}}) \right) \tag{4.8}$$

where positive factors λ_i control the weighting of respective high-level application metrics M_i (assuming larger scores to be better) and proxy functions $f_i, g_i \colon \mathbb{R} \to \mathbb{R}$ to ensure linear relationships $\alpha \propto f_i \propto g_i$. This extension does not increase the computational

cost at runtime, since $\alpha^\star$ can be calculated offline for each bin of blur/ noise levels and stored in look-up-tables.

Generality

The underlying assumptions of the proposed decision & control unit are stated in Sec. 4.4.2: (i) the produced image intensity I is proportional to the exposure time t_{exp}, (ii) camera sensor linearity, (iii) constant speed between camera and scene, and (iv) digital ISO gain. A different relationship for (i) would result in changed relations in (4.5) and hence a different calculation of α in (4.6). The relations in (4.5) require only that t_{exp} is a function of I and that the relationship between the two is known (e.g., empirically determined). Likewise, the same relations would change for (ii) non-linear camera sensors (such as sensors with activated gamma correction post-processing or logarithmic response CISs [Kav+00]) and (iv) a non-digital-only ISO gain (e.g., a combination of analog and digital gains). If the (iii) relative speed between the camera system and scene objects cannot be kept approximately constant in the case of occurring motion blur, the framework may trigger camera parameter changes repeatedly within a short time. An example could be an autonomous driving scenario with oncoming traffic, where vehicles pass faster than the camera parameters can be adjusted. If this behavior is undesired, approaching vehicles in the oncoming lane could be neglected in the framework analysis (e.g., through lane detection and filtering).

4.6 Summary

This chapter introduced our proposed camera self-health-maintenance framework for autonomous mobile machines.

Section 4.1 briefly recapitulated the objectives and the scope of the framework, and Sec. 4.2 provided a high-level overview of the framework's offline training and online testing parts, and its basic components: the condition estimation and the decision & control units.

Subsequently, Sec. 4.3 first introduced the condition estimation unit. There we distinguished between two cases: total blur/ noise cases and multiple cases. For the first case, we presented improved learning-based total blur/ noise estimators for the framework and traditional approaches for evaluation. On this basis, we further proposed a novel noise source estimation approach that combines the learning-base and the physical-based paradigms.

In addition, Sec. 4.4 presented the decision & control unit. We first considered the empirical sensitivity analysis of object detection performance with respect to blur and noise to form input-output performance curves. These curves were then employed to

determine camera parameter adjustments on the basis of online noise/ blur estimations in order to optimize object detection performance.

Finally, we briefly discussed the framework design (Sec. 4.5). First, limitations of the suggested estimators and an approach for blur source estimation were presented. Next, we considered extensions for the decision & control unit, specifically, for additional image quality attributes and multiple target applications. Lastly, the general applicability of both units and different Sensor AI approaches were addressed.

In the following chapters, we evaluate the reliability of the proposed blur and noise estimators (Ch. 5), and the noise source estimator (Ch. 6). The practical application of the framework on mobile hardware and its analysis in terms of the required computational cost are subsequently evaluated in Ch. 7.

CHAPTER 5

Evaluation: Blur and Noise Estimation

This chapter covers the evaluation of the proposed learning-based and traditional blur and noise estimators from the framework's condition monitoring module. We first introduce employed synthetic and real-world corrupted datasets (Sec. 5.1). Subsequently, we evaluate blur and noise estimators on respective isolated corruptions (Secs. 5.2 and 5.3), and on simultaneously occurring corruptions (Sec. 5.4). On this basis, we propose two improvements: blur estimation in the presence of high noise (Sec. 5.5) and noise estimation with reduced uncertainty (Sec. 5.6). Finally, limitations and further potential improvements are discussed (Sec. 5.7) and the chapter is summarized (Sec. 5.8). This chapter is partially published in [Wis+23b].

All experiments are executed on an Intel Xeon W-2145 CPU and an NVIDIA Quadro RTX 6000 GPU, with the neural networks running on the GPU.

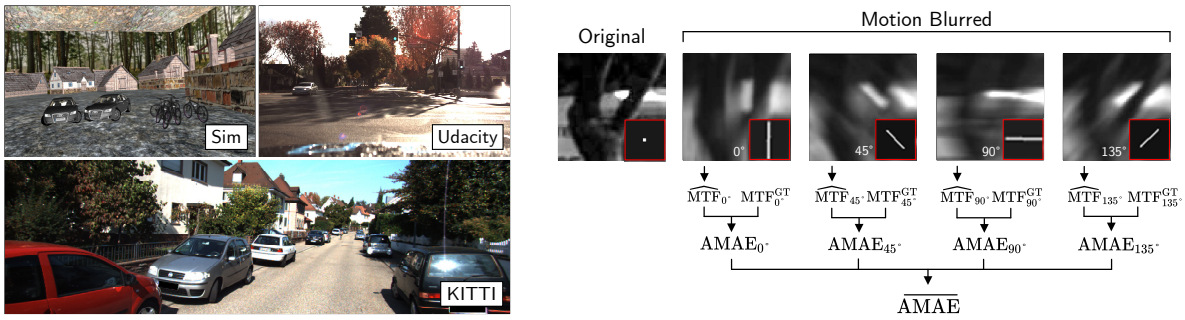
5.1 Datasets

The proposed noise and blur estimators are evaluated on five datasets: three with simulated blur and noise corruptions (Sec. 5.1.1), and two with real-world defocus and motion blur, respectively (Sec. 5.1.2). Note that real-world noise evaluations are part of Ch. 6.

5.1.1 Simulated Corruptions

We employ one simulated and two real-world datasets: *Sim*, KITTI [GLU12] and Udacity [Uda16] (Fig. 5.1a).

We create *Sim* with the simulator [Irm+19] to provide accurate ground truth for blur and noise estimation. *Sim* comprises 1000 images of a village environment acquired from different viewpoints and includes vehicles, such as cars and bikes. From KITTI we use the annotated object detection sub-dataset (with preceding frames), and from Udacity we use sub-dataset #2. We subsample KITTI and Udacity for two reasons:



(a) Datasets **(b)** Motion Blur Kernel Rotation

Figure 5.1: *Datasets with simulated corruptions.* (a): Exemplary images from datasets Sim (896×768 px), Udacity (1920×1200 px) and KITTI (1242×375 px). (b): We mitigate the influence of motion blur direction by evaluating four rotated versions of each kernel separately. For each image and each kernel version, we calculate the average mean absolute error (AMAE, defined in (5.1)) from the estimated blur ($\widehat{\mathrm{MTF}}$) and the associated ground truth label ($\mathrm{MTF}^{\mathrm{GT}}$), and use their average ($\overline{\mathrm{AMAE}}$) as final result.

to reduce processing time and to remove (in all conscience) clearly visible blur/noise corrupted images that would bias estimation results. To this end, we pick 1000 images per dataset for noise estimation and 150 images for blur estimation, and match these numbers on *Sim*. For blur, we only use image patches containing detected objects of interest. However, a residual risk of corruption in the natural images remains, so we further reject $\approx 5\,\%$ outliers with respect to the blur and noise estimation error scores in the experiments (using the modified z-score with default values from [IH93, pp. 11–13]).

All datasets are synthetically corrupted with controlled amounts of noise and blur using the models of Sec. 3.2.

Noise

Following the "real noise" studies in [AB18], we generate noise with levels $\sigma \in \{5, 10, 15, 20, 25\}$ DN. We apply default CMOS camera parameters from [KW14] and study noise (*i*) in isolation or (*ii*) in combination. (*i*) For isolated DCSN and readout noise studies, we set the temperature to $T = 330\,\mathrm{K}$ and the exposure time to $t_{\mathrm{exp}} = 0.1\,\mathrm{s}$. (*ii*) For the combined noise case, we include *all* noise sources with random $T \in [300,\, 330]\,\mathrm{K}$ and $t_{\mathrm{exp}} \in [0.002,\, 1]\,\mathrm{s}$ to emphasize different noise components in each image. In order to reach the desired σ, we amplify the (raw) noise in both settings.

The datasets yield the following number of non-overlapping image patches (128×128 px) per noise level: 42k (Sim), 18k (KITTI), and 135k (Udacity).

Blur

We synthesize blur kernels of size $d \in \{3, 7, 11, 15, 21\}$ px. d is the diameter for defocus kernels or the approximate path length for motion blur kernels. Defocus blur kernels are calculated analytically. Motion blur kernels are generated using [Bor20], distinguishing between linear motion kernels (motion intensity parameter of [Bor20] set to 0) and

non-linear ones (parameter set to 1.0), and manually selecting the kernels that satisfy the target d. We mitigate the influence of motion blur direction by evaluating four rotated versions of each kernel separately (rotating them with angle $\alpha \in \{0, 45, 90, 135\}^\circ$ counterclockwise, see Fig. 5.1b).

All datasets provide about 600 non-overlapping image patches (192×192 px) per investigated blur kernel.

Combined Blur and Noise

We further propose two use cases for *combined blur and noise* occurrences: (*i*) *Defocus blur and DCSN* (Defocus + DCSN) that might arise at high temperatures (as caused by direct Sun illumination) and with defocus induced by material stress in the optics setup [Küh+20; KBE21], and (*ii*) *photon noise and motion blur* (Photon + Motion) due to high exposure times and signal amplification, typical of low light conditions.

We analyze the same image patches as for the isolated blur and noise cases, respectively.

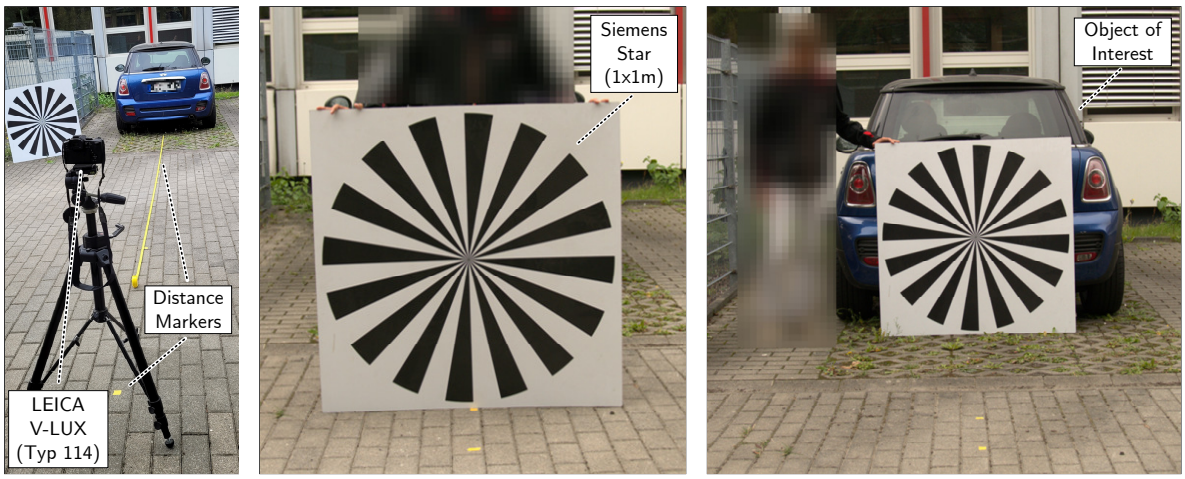
5.1.2 Real-World Corruptions

In addition to the synthetically corrupted datasets, we propose two self-created image datasets with real-world defocus and motion blur, respectively. The datasets are referred to as DEFCARS ("DEFocused CARS") and MOTCARS ("MOTion blurred CARS").

Defocus Blur

The DEFCARS dataset contains 104 images of three different cars in an open parking lot. We target to create the same defocus blur kernel sizes as for the synthetically blurred datasets (cf. Sec. 5.1.1). To this end, we vary the camera focus distance d_O and fix the other defocus blur parameters specified in the defocus blur model (3.7). In order to induce the kernel sizes within manageable distances d_O (the parking lot scene allows a maximum of $d_O = 6\,\text{m}$), we use a *LEICA V-LUX Typ 114* [Lei16] camera with a lens system that provides a wide range of adjustable focal lengths f to trade-off d_O with f for a fixed kernel size d. With the camera's pixel pitch $d_p = 2.4\,\mu\text{m}$, an aperture diameter $D_A = 6.03\,\text{mm}$, and an out-of-focus object distance $d_B = 6\,\text{m}$, we calculate a focal length $f = 21.11\,\text{mm}$ and approximate focused object distances $d_O \in \{6.0, 4.5, 3.4, 2.7, 2.2, 1.8\}\,\text{m}$ from (3.7) to produce the desired blur kernel sizes d. Tab. A.1 provides details to the LEICA camera system.

Figure 5.2 summarizes the image acquisition procedure. First, we measure the distances d_O and d_B between the center of the camera lens system and a car's number plate, and position the camera on a tripod at d_B (Fig. 5.2a). Second, we place the Siemens star at a distance d_O and focus the camera to it by manually triggering the camera's auto-focus (Fig. 5.2b). Third, we keep the camera focus and image the car with the Siemens star



(a) Experim. Setup **(b)** Adjust Focus Distance **(c)** Determine Defocus Blur

Figure 5.2: *Real-world defocus blur dataset DEFCARS.* (a): Camera system on a tripod and distances that correspond to different blur kernel sizes measured from the object of interest (car). (b): Siemens star used as target to manually focus the camera to a specific distance. (c): With camera focus set, the object of interest is recorded with the Siemens star on it to determine the ground truth defocus blur.

on it (Fig. 5.2c). The Siemens star is used to determine the ground truth defocus blur. We repeat this procedure for each measured d_O and finally obtain 52 images (about 4k car image patches) for blur estimation and 52 images for camera focusing. For each image acquisition, the camera height is adjusted so that the center of the image aligns with the center of the Siemens star or the center of the license plate, respectively. All images are evaluated in raw grayscale format.

The top row of Fig. 5.3 depicts our defocus blur ground truth determination in five steps: (1) In case the unblurred imaged Siemens star center is $\approx 1\,\text{px}$ large, the resulting CoC of the blurred Siemens star can be approximated visually. (2) Since this procedure is subjective and thus inaccurate, we provide a minimum and a maximum CoC with its diameters as blur kernel sizes. (3) Next, we apply the Siemens star image to the resolving power tool and determine the ground truth MTF^{GT} using the Siemens star method (Sec. 3.3.1.2). (4) The PSF can be directly derived from the MTF^{GT} by plotting the reciprocal MTF value at the Nyquist frequency for each analyzed radius from the Siemens star center (assuming a Gaussian-shaped and a rotationally symmetric PSF) [Mei+20]. (5) To verify the results of step 2, minimum and maximum blur kernel diameters are further estimated on the PSF (the resolving power tool provides an interactive plot in pixel units). We only keep images where the mean blur kernel sizes from steps 2 and 5 differ by at most 10% or 1 px (whichever is larger).

Motion Blur

The MOTCARS dataset contains 25 carefully selected raw grayscale images of three cars (75 car image patches) in a parking garage recorded with an *Allied Vision Prosilica*

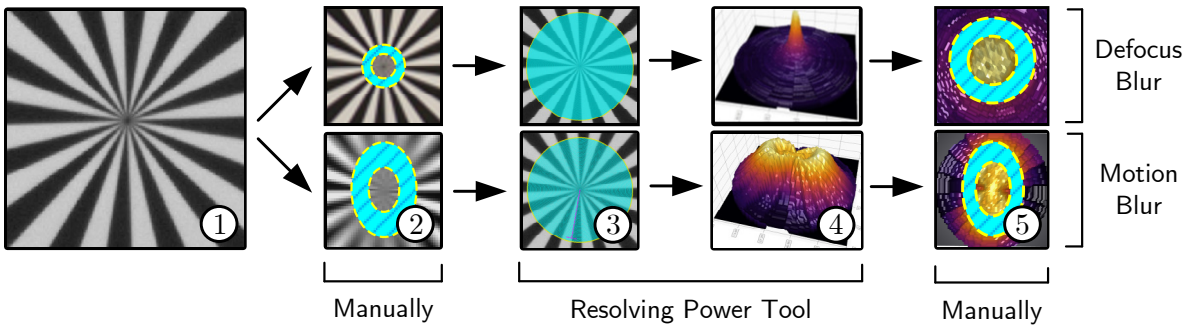
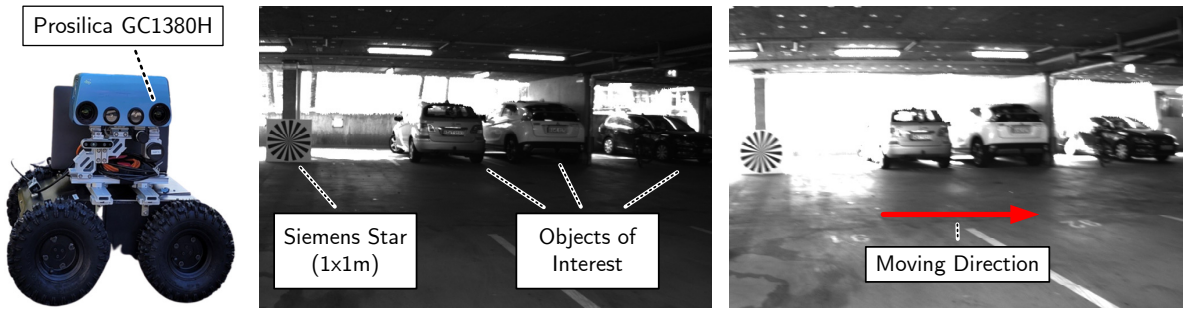


Figure 5.3: *Ground truth determination of real-world defocus blur (top row) and motion blur (bottom row).* (1): Siemens star image acquisition. (2): Manual minimum/maximum blur kernel size estimation (circle diameter for defocus blur and ellipsoid length minus its width in the case of motion blur). (3): Ground truth $\mathrm{MTF}^{\mathrm{GT}}$ determination with Siemens star method or SLE method, respectively, using the resolving power tool. (4): PSF approximation from $\mathrm{MTF}^{\mathrm{GT}}$. (5): Verification of results from step (2) on the PSF.



(a) Cam on Robot **(b)** Experimental Setup **(c)** Induce Motion Blur

Figure 5.4: *Real-world motion blur dataset MOTCARS.* (a): Camera system mounted on a moving robotic platform. (b): Siemens star positioned alongside the cars of interest for ground truth motion blur determination. (c): Robot moves parallel to cars with different exposure times to induce motion blur.

GC1380H camera [Gmb21] attached on a Jaguar-4x4-wheel mobile robotic platform [Dr 21] (Fig. 5.4a). The robotic platform enables a controllable constant movement to create realistic motion blur in mainly horizontal image direction (Fig. 5.4c). The camera uses a fixed-focus lens system (details in Tab. A.1) and is focused on its hyperfocal distance of $H \approx 1.43\,\mathrm{m}$ (3.9) to aim for defocus blur of $d \leq 1\,\mathrm{px}$ for distances beyond $H/2$. In order to induce different motion blur sizes, we set the camera exposure time to $t_{\mathrm{exp}} \in \{4, 6, 7, 8, 10\}\,\mathrm{ms}$ and fix other camera parameters. These exposure times are selected manually to avoid excessive under- and overexposure in the images. Note that images with higher exposure times are not necessarily brighter, as the lighting conditions change between the image acquisitions.

Analogously to the DEFCARS dataset, we image a Siemens star for ground truth determination. Specifically, we place a Siemens star at the same distance as the cars (Fig. 5.4b), move the robot with a constant speed parallel to the cars, and record images of the scene. This procedure is repeated with the different t_{exp}.

The ground truth motion blur determination follows the same approach as for DEFCARS, with two changes (bottom row in Fig. 5.3): In steps 2 and 5, we use an ellipsoid to fit the Siemens star center blur [Mei+20]. The ellipsoid expansion h in motion direction includes both the defocus size d_{defocus} and the motion blur size d_{motion} with $h = d_{\text{defocus}} + d_{\text{motion}}$. The ellipsoid size w perpendicular to the motion direction contains only defocus blur (i.e., $w = d_{\text{defocus}}$). Hence, $d_{\text{motion}} = h - w$. The second change is that we apply the SLE method within the resolving power tool in step 3 (see Sec. 3.3.1.1). For more robustness, we analyze two opposite edges of each Siemens star at about $\theta \in \{5, 185\}\,^\circ$ with respect to the image's vertical axis and calculate their mean MTF in horizontal image direction.

5.2 Blur Estimation

We assess blur estimation accuracy in terms of the *average mean absolute error* (AMAE) between a median MTF estimation ($\widetilde{\text{MTF}}$) and ground truth (GT) samples at eight frequencies (f_i) each in horizontal (H) and vertical (V) image directions (w):

$$\begin{aligned} \text{AMAE} &\doteq \frac{1}{2} \sum_{w=\{\text{H,V}\}} \text{MAE}(w), \\ \text{MAE}\,(w) &\doteq \frac{1}{8} \sum_{i=1}^{8} \left| \text{MTF}_w^{\text{GT}}(f_i) - \widetilde{\text{MTF}}_w(f_i) \right|. \end{aligned} \tag{5.1}$$

The motivation behind this metric is addressed in Sec. 5.7.

We further consider the variance of the median MTF estimation (in this context, the range between minimum and maximum MTF estimations) as an indicator of robustness or uncertainty, respectively.

5.2.1 Simulated Blur

Let us first apply the blur estimators to the uncorrupted datasets, subsequently examine accuracy and robustness using the corrupted datasets, and finally their computational performances.

Uncorrupted Datasets

Median, minimum and maximum estimations (rejecting $\approx 5\%$ outliers) are first calculated for the uncorrupted datasets in Fig. 5.5. In the *Sim* case, we determine MTF^{GT} by evaluating a Siemens star (generated in the simulator) with the resolving power tool [Mei20]. For the real-world datasets however, there are no known GT values, but we expect similar sharp images and hence we plot the estimations for comparison.

Analyzing Fig. 5.5 we make five major observations: (i) The CNN estimates a nearly ideal MTF with hardly any variance in the *Sim* case and provides similarly confident

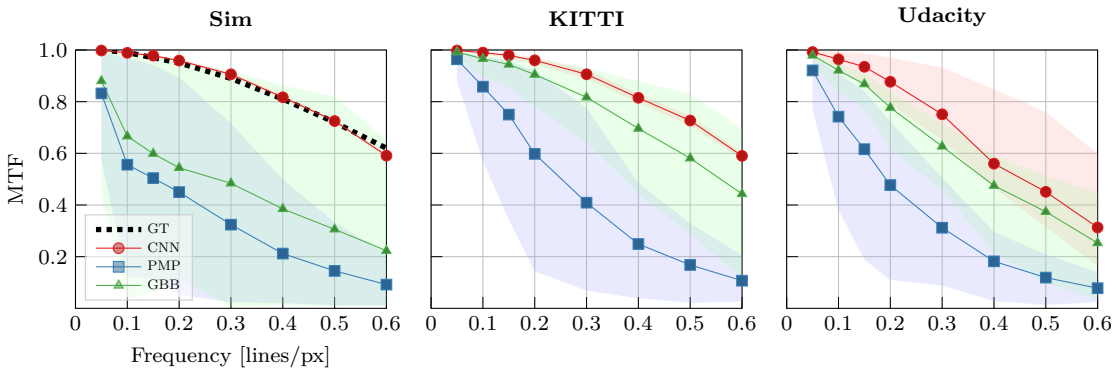


Figure 5.5: *Blur estimation of uncorrupted datasets (i.e., "ground truth").* Median, minimum and maximum blur estimations of the uncorrupted datasets (depicted by sampled points with interpolation in between and the shaded areas, respectively; horizontal direction only).

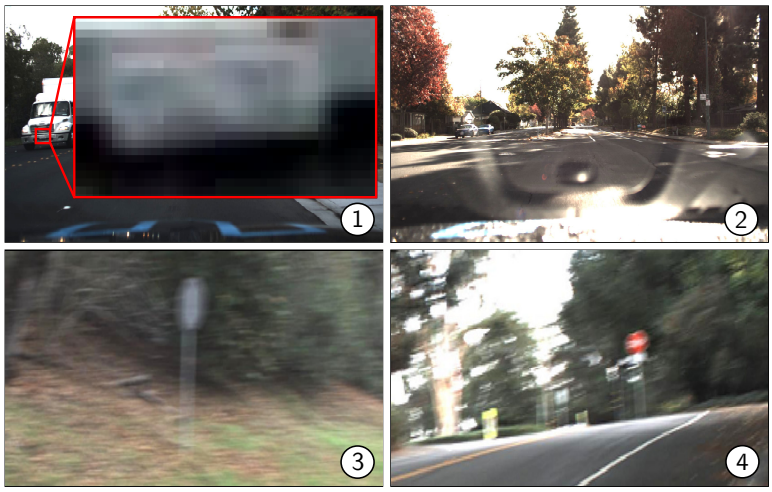
estimations for KITTI. (*ii*) Contrary to expectations, the CNN estimates a more uncertain (less robust) and lower MTF for Udacity. Concerning this, we found challenging effects that influenced the estimation, like frequent windshield reflections and regular slight motion blur in the moving direction, despite our pre-selection of images. The traditional estimators (GBB/PMP) are also affected, producing lower median estimations than for KITTI. (*iii*) The variances of GBB/PMP shrink from *Sim*, via KITTI towards Udacity. (*iv*) GBB performs noticeably worse in *Sim*. We ascribe its low median and large variance to the lack of image gradient diversity of the *Sim* dataset (GBB relies on gradients, but strong horizontal edges are scarce in *Sim*). (*v*) PMP produces generally low estimations and its maxima are far from the GT (*Sim*) or expected GT (real-world) values.

Corrupted Datasets: Accuracy

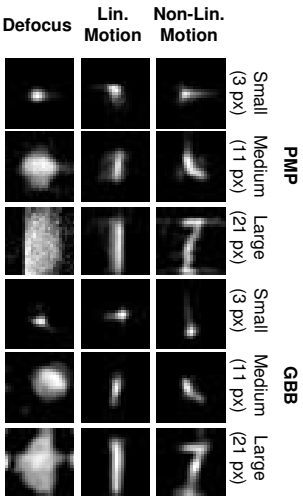
Next, we corrupted the datasets with the generated blur kernels and used the sampled MTFs of the kernels as ground truth. The blur AMAE scores are summarized in Tab. 5.1. We make the general observation that PMP and GBB —unlike the CNN— usually perform worst for small (3 px) and large (21 px) kernel sizes, respectively. This often manifests in undesired artifacts like smear or cuttings in these estimations (Fig. 5.6b). The decreased performance for small blur cases is in agreement with the results from Fig. 5.5, where no additional blurring was added. There, GBB and particularly PMP produce lower median estimations and higher variance for *Sim*/KITTI, and lower variance for the already corrupted Udacity. Since GBB/PMP follow a coarse-to-fine approach, more internal iterations would enhance the level of detail of the kernel and thus produce smaller errors (at the expense of computational cost). On the other hand, larger kernel estimations improve as larger image patches are used. Therefore, the authors of GBB [Bai+18] suggest kernels to be much smaller than the image to have a well-defined blur estimation problem. We further regularly observe larger estimation

		Defocus Blur					Linear Motion Blur					Non-linear Motion Blur				
Size [px]		3	7	11	15	21	3	7	11	15	21	3	7	11	15	21
Kernel																
Sim	CNN	**0.7**	**1.8**	**2.1**	**0.5**	**1.1**	**6.3**	**10.3**	9.4	9.4	**7.7**	**2.9**	12.2	11.4	19.5	25.0
	PMP	13.9	5.2	3.0	5.3	6.7	37.2	17.8	13.3	**7.3**	16.6	21.8	14.0	13.8	9.8	**11.5**
	GBB	2.7	3.8	6.3	8.4	17.6	31.6	11.4	**7.7**	**7.3**	14.5	17.7	**8.3**	**8.2**	**9.7**	15.0
KITTI	CNN	**0.3**	5.3	2.7	**2.3**	**0.7**	**3.4**	10.9	9.9	9.1	**6.6**	**4.0**	14.1	11.2	14.1	9.2
	PMP	5.8	**2.3**	**1.5**	2.9	4.8	37.2	12.2	8.7	**4.1**	9.5	22.5	7.9	7.3	5.2	4.2
	GBB	3.2	2.9	2.6	**2.3**	9.3	13.4	**5.8**	**5.3**	4.7	7.1	8.5	**4.0**	**5.2**	**3.5**	**3.7**
Udacity	CNN	**2.7**	**0.9**	**0.6**	**0.3**	**1.4**	**16.2**	**10.8**	9.8	11.4	**7.9**	**9.6**	10.3	11.9	16.0	19.2
	PMP	15.1	5.6	4.0	3.9	3.9	34.3	14.2	11.5	**8.1**	12.4	23.5	12.0	**11.0**	**8.1**	**7.6**
	GBB	2.8	8.8	8.6	8.9	23.2	21.7	12.2	**8.5**	12.2	21.1	10.6	**10.1**	11.6	13.5	16.6

Table 5.1: *Blur estimation of synthetically corrupted datasets.* Ground truth blur kernels and average mean absolute errors (AMAE) of horizontal and vertical median blur estimations [%]. The best results per kernel and dataset are highlighted in bold.



(a) Challenging conditions in Udacity dataset



(b) Undesired artifacts in kernel estimations

Figure 5.6: *Undesired artifacts in blur estimation.* (a): Slight motion blur of 3 px in moving direction (1), light reflections (2) and two examples of severe motion blur (3, 4). (b): Typical GBB/PMP kernel estimations with undesired artifacts (compare to respective ground truth kernels from Tab. 5.1).

errors for Udacity. This confirms that Udacity is already corrupted by blur and/or the estimations are influenced by challenging conditions (Fig. 5.6).

Apart from the already mentioned small/large kernels, all methods estimate defocus well (Tab. 5.1). Nevertheless, the CNN delivers the most accurate results. GBB considers the common simplification of Gaussian blur for defocus, whereas PMP does not and tends to perform slightly better than GBB.

The CNN also estimates linear motion blur comparably well but (except for small/large kernels) GBB tends to produce the smallest errors.

Non-linear motion estimation results (also in Tab. 5.1) differ for the CNN method, which tends to produce larger errors towards the larger (and more complex) kernels compared to the traditional estimators and the linear case. We interpret this as a larger

uncertainty and conclude that the CNN might not be appropriate for estimation of complex non-linear motion kernels. In contrast, the scores of GBB/PMP are more accurate among the different kernels and datasets (with GBB a bit better). This slightly better motion blur estimation performance of GBB compared to PMP is consistent with the experiments in [Bai+18], where PMP is compared to the work of Pan et al. [Pan+17a] that first proposes a dark channel prior.

Corrupted Datasets: Robustness

The MTF graphs associated with Tab. 5.1 are shown in Figs. B.1, B.2, and B.3 (for the sake of clarity and readability, these are located in the appendix), which we use to assess the robustness of the blur estimators.

For all blur types, the first general observation is that if an estimator produces results with low variance, it also shows high accuracy. It can further be seen that PMP and GBB are most uncertain for small kernel sizes $d = 3\,\text{px}$ (PMP and GBB) and large sizes $d = 21\,\text{px}$ (GBB only), which is consistent with the observations on their low accuracies.

In the case of defocus blur (Fig. B.1), the CNN shows almost no variance, corresponding to its high accuracy (except for Udacity $d = 3\,\text{px}$ with its initial blur, cf. previous findings on accuracy). Comparing the three methods, GBB shows the highest variance, followed by PMP.

When it comes to linear motion blur (Fig. B.2), the CNN and PMP variances tend to be higher for less accurate estimates. However, GBB shows mismatches between accuracy and variance (higher accuracy with higher variance, for instance, in *Sim* ($d = 21\,\text{px}$), KITTI ($d \in \{11, 21\}\,\text{px}$), and Udacity ($d \in \{11, 21\}\,\text{px}$)). Apart from the small and large kernel cases, PMP produces the lowest variance, followed by GBB and CNN.

For non-linear motion blur (Fig. B.3), CNN shows higher variance for the more complex kernels ($d > 3\,\text{px}$), which agrees with the previous finding that the CNN is not suitable for estimating complex non-linear kernels. At this point, CNN's variances do not match its accuracies. The same is true for GBB, whose variances tend to increase with increasing kernel size ($d \geq 11\,\text{px}$ for *Sim* and for $d \geq 15\,\text{px}$ otherwise). In contrast, PMP has relatively low variances for $d > 3\,\text{px}$, corresponding to its higher accuracies.

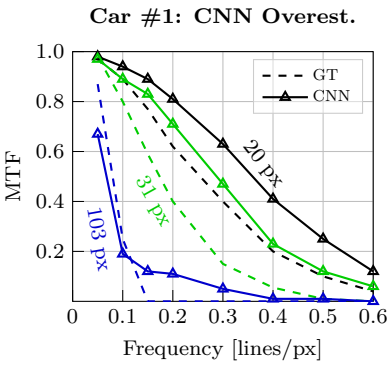
All results show that larger uncertainties are typically associated with lower accuracy. In particular, no estimator yield high certainty at low accuracy. For this reason, we recommend to evaluate the variance together with the median for short time intervals as an indicator of the expected accuracy of an estimator (assuming constant blurring within this time interval).

Computational performance

In order to assess the real-time capability of each blur estimator, we perform intermediate

		Defocus Blur					
Size [px]		20–25	25–30	30–35	35–40	50–55	>100
Car #1	CNN	12.9/13.0	12.6	16.5	-	-	**7.0**
	PMP	**11.5/10.4**	**10.5**	**12.9**	-	-	11.1
	GBB	13.9/13.1	16.0	14.8	-	-	12.5
Car #2	CNN	-	9.3	10.5/6.3/7.8	10.1	-	**5.4**
	PMP	-	4.3	2.9/3.9/3.6	5.9	-	8.8
	GBB	-	**3.1**	**1.5/3.5/3.3**	**4.9**	-	12.9
Car #3	CNN	-	11.4	9.0/5.3	3.8	**4.9**	**5.4**
	PMP	-	6.5	3.9/3.4	3.8	7.6	11.0
	GBB	-	**5.1**	**2.6/4.0**	**3.5**	7.4	12.3

(a) DEFCARS defocus blur estimation results



(b) CNN overestimation

Figure 5.7: *Defocus blur estimation on real-world corrupted DEFCARS dataset.* (a): Average mean absolute errors (AMAE in %). The best results per blur size bin and object (car) are highlighted in bold. (b): CNN overestimates real-world defocus blur in DEFCARS data.

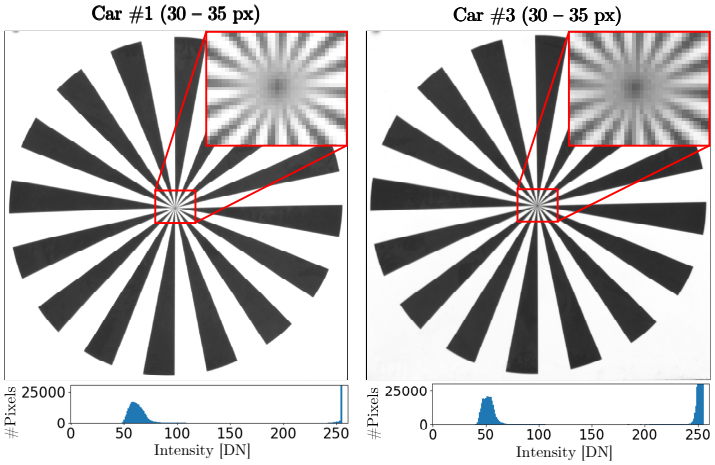
runtime measurements at this point; a more comprehensive analysis is provided in Ch. 7. We measure the following mean runtimes: 13.07 s (GBB, per image patch), 12.69 s (PMP, per image patch), and 0.24 s (CNN, per input batch of four images). The CNN executes more than ×50 faster than GBB/PMP and moves in the realm of real-time capability. We also found that CNN requires 98% of its runtime for serial data pre-processing, which can be improved by vectorization (details in Sec. 7.4). Although the CNN itself executes on a GPU, the running times of current GBB/PMP implementations (running on the CPU) are too long to be practical for a condition monitoring application (especially for multiple image patches).

Summary

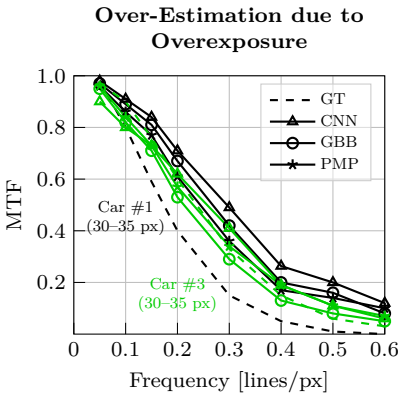
In summary, the GBB and PMP methods are in general neither accurate nor robust for blur-free or small/large blur kernel estimation on the image patch sizes used, and available implementations are not real-time capable. Nevertheless, they provide the best estimates for medium-sized linear and non-linear motion blur kernels. The CNN method, on the other hand, might not be suited for complex non-linear motion kernels, but performs well in terms of defocus, linear motion and real-time requirements. If non-linear motion blur can be circumvented (e.g., with short exposure times or slow motions), the CNN method can be employed for monitoring a camera's condition. For more robust estimates, we recommend using median statistics over small time spans and the variance to assess estimation uncertainty.

5.2.2 Real-World Defocus Blur

The real-world defocus blur estimation results are summarized in Fig. 5.7 and reported as non-aggregated AMAE values per image. Supplements are provided in App. B.1.2 (Fig. B.4).



(a) Overexposed vs. well-exposed Siemens star



(b) Corresp. MTF estimations

Figure 5.8: *Overestimation of defocus blur in overexposed images.* (a): Exemplary image crops acquired from car #1 and car #3 scenes with a defocus blur size of $d \in [30, 35]$ px. The car #1 image is strongly overexposed (see corresponding histogram below with a peak at 255 DN), the car #3 image is less overexposed by comparison. (b): Corresponding MTF estimations. Missing intensity values due to overexposure indicate a lower degree of blur, i.e., higher MTF values, which leads to overestimation.

We first note that the visually determined ground truth kernel sizes are significantly larger than the calculated ones provided by the theoretical model (cf. Sec. 5.1.2 with Fig. 5.7b). These differences are mainly caused by a model deviation from reality and measurement inaccuracies. The model deviation is in agreement with the results of Seo [Seo20], who compared the blur from the thin lens model to a similar camera system (Nikon D300S [Nik23]) and stated that "*[...] the discrepancy between the theoretical blur amount and the blur amount of the DSLR camera was found to be non-trivial*". However, the reasons are not identified. Potential lens system model deviations and further reading are provided in Sec. 3.4.1. Measurement errors have the greatest relative impact on the lens system's side, since a small deviation from the true optical center of a lens system results in different d_I and d_O in (3.7). As opposed to a single lens, the optical center of a multi-lens system can differ from the center of the lens system we assumed [Hec17, p. 169, pp. 178–181]. To cluster our defocus blur estimation results, we bin the kernel sizes into six categories.

When it comes to blur estimation accuracy, we make four major observations on Fig. 5.7: (*i*) Although the CNN is trained on defocus blur sizes $d \leq 21$ px (cf. Sec. 4.3.1), its estimation accuracy is comparable to the ones of GBB and PMP, suggesting that CNN learned to generalize for $d > 21$ px. (*ii*) CNN generally produces higher AMAE scores compared to GBB/PMP and to the results on synthetically corrupted datasets (Sec. 5.2.1). This is due to the original supervised CNN training, where the CNN only learned the corruption blur, but not the initial blur in the uncorrupted images that is introduced by the camera system (cf. [Bau+18]). This makes the CNN prone to overestimation (Fig. 5.7b). Note that this additional camera blur is part of the

determined real-world defocus ground truth but not part of the ground truth of the synthetically blurred datasets. (*iii*) Estimation errors of GBB and PMP are comparable to those of the simulated experiments, but increase only for $d \geq 50\,\mathrm{px}$, while they increase for $d = 21\,\mathrm{px}$ in simulated blur experiments. This supposedly overall larger kernel sizes in DEFCARS result from the camera sensor's smaller pixel pitch[1]. In order to compare a blur size $d^{(c1)}$ with $d^{(c2)}$ from two camera systems $c1$ and $c2$, the following transformation has to be applied (cf. (3.7)):

$$d'^{(c1)} = \frac{d^{(c1)} d_p^{(c1)}}{d_p^{(c2)}}, \tag{5.2}$$

with $d'^{(c1)}$ being the blur size of $c1$ in the coordinate system of $c2$ and the respective pixel pitches $d_p^{(c1)}$ and $d_p^{(c2)}$. That is, a blur size of, for instance, 21 px in KITTI data acquired with a camera sensor having $d_p = 4.65\,\mu\mathrm{m}$ [GLU12; Aud23] would be comparable to a DEFCARS blur size of $\approx 41\,\mathrm{px}$, which is close to where the GBB and PMP blur estimation errors start to increase in Fig. 5.7a. (*iv*) All estimators yield higher estimation errors for images of car #1. That is because these images are overexposed (Fig. 5.8a), which was not mitigated by the camera's default automatic exposure control. Overexposure results in a loss of grayscale values generated by the blur process, which are therefore important for estimating blur. In turn, the missing grayscale values indicate a lower degree of blur, i.e., higher MTF values, which were produced by all estimators (Fig. 5.8b). We expect the same effect for underexposed images.

Summary

In real-world defocus estimation, CNN performs generally less accurate than GBB and PMP as it does not account for initial blur of a camera system. This error can be reduced in a re-training to enable a camera-agnostic monitoring. However, PMP and GBB are still inferior to CNN in estimating large defocus blur sizes. The experiment also reveals that all estimators are susceptible to overexposure in that they estimate lower blur. Therefore, image intensity should also be taken into account for reliable blur estimation in a condition monitoring.

5.2.3 Real-World Motion Blur

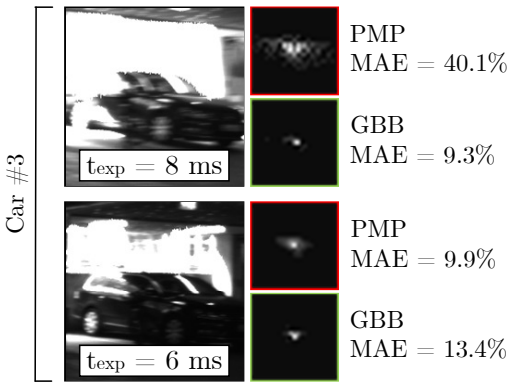
Real-world motion blur estimation results are summarized in Fig. 5.9 and supplemented by the appendix Fig. B.4. Results are reported in median MAE scores for the horizontal image direction in binned categories for the determined motion blur sizes.

We notice three aspects: (*i*) All blur estimators generally produce higher errors than in the simulated blur experiments (Sec. 5.2.1). This is in accordance with the real-world defocus blur results for car #1 and comes from severe over- and underexposure

[1] Note that our MTFs are specified in pixel units and are thus relative to the pixel pitch (cf. Sec. 3.3.1).

Size [px]		Motion Blur 1–4	4–6	6–10	12–16	16–20
Car #1	CNN	11.9	16.4	**10.6**	22.1	21.7
	PMP	30.7	23.8	25.3	**15.4**	18.5
	GBB	**7.4**	**16.3**	14.9	32.3	**12.1**
Car #2	CNN	**6.3**	**5.7**	9.3	19.2	13.6
	PMP	41.5	29.7	25.1	15.8	26.4
	GBB	9.9	9.9	**6.4**	**5.6**	**6.4**
Car #3	CNN	12.5	**10.8**	**10.0**	19.4	**17.9**
	PMP	34.5	23.8	20.0	10.1	18.1
	GBB	**3.3**	15.0	11.4	**18.9**	19.1

(a) MOTCARS motion blur estimation results



(b) Influence of overexposed areas

Figure 5.9: *Motion blur estimation of real-world corrupted datasets.* (a): Median mean absolute errors (MAE (5.1)) of horizontal blur estimations [%]. The best results per motion blur size bin and object (car) are highlighted in bold. (b): Exemplary image patches of car #3 taken with exposure times $t_{\text{exp}} \in \{6, 8\}$ ms and corresponding blur kernel estimations of PMP and GBB. In case of $t_{\text{exp}} = 8$ ms, severe overexposure reduces image information, which prevents accurate PMP blur estimation (undesired artifacts in corresponding blur kernel and high MAE of 40.1%). In case of $t_{\text{exp}} = 6$ ms, the less overexposed image results in better PMP blur estimation (smooth blur kernel shape and lower MAE of 9.9%). In contrast, GBB blur estimations are less prone to image content loss.

in the images (see Fig. 5.4 and compare to Fig. 5.8). Each image of MOTCARS contains over- and underexposed areas near the car objects, since the used camera system is not able to capture the high dynamic range of the scene (which is a common issue of conventional cameras [Reb+19]). Moreover, the CNN is further affected by the aforementioned influence of the initial camera system blur (cf. Sec. 5.2.2). (*ii*) Analogous to the simulated and real-world defocus experiments, all methods are prone to estimation errors for small or large kernel sizes. CNN produces noticeably higher errors for $d \geq 12$ px (similar to the synthetic non-linear blur kernel results), PMP for small kernels ($d \in [1, 4]$ px), and GBB for three large kernels ($d \geq 12$ px). (*iii*) PMP produces significantly higher estimation errors compared to the synthetic blur experiments. This effect comes from the reduced information in images with large overexposed areas and is illustrated in Fig. 5.9b (note that unlike Fig. 5.8, not the entire image is overexposed, but only areas near the outside of the car park). The figure demonstrates that PMP estimates a blur kernel with undesired artifacts in the case of severe overexposure (which leads to a high MAE score of 40.1%) and a smoother kernel in a comparable image with less overexposure (which leads to a lower MEA score of 9.9%). We observe such kernel artifacts also in the simulated blur experiments for small/large blur kernels (cf. Fig. 5.6b), with the same underlying reason of an ill-posed blur estimation problem (Sec. 5.2.1). In comparison, GBB is less susceptible to this level of information loss.

Summary

In summary, the results support the finding of the previous experiments that the estimators are inaccurate for small (PMP) or large (CNN and GBB) motion blur.

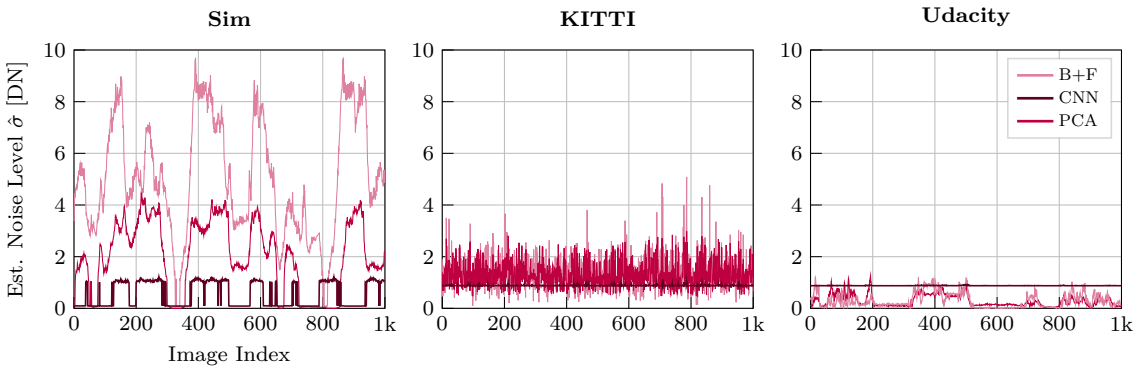


Figure 5.10: *Noise estimation of uncorrupted datasets.* All reference methods estimate little median noise in KITTI and Udacity images ($\tilde{\sigma} \lessapprox 1.0\,\mathrm{DN}$). However, PCA and B+F incorrectly estimate significant noise in Sim images (with $\hat{\sigma}$ often in the range of $[2, 10]\,\mathrm{DN}$).

The results also show that all estimators are affected by under- and overexposed areas, which reduce the information content of an image that is vital for blur estimation; especially PMP becomes impractical in the case of large overexposed image areas. Nevertheless, CNN and GBB provide the best estimates for small and medium-sized motion blur kernels. This experiment confirms the necessity to avoid or minimize under- and overexposure for a reliable blur estimation (e.g., by using an additional under-/overexposure estimator or by a custom exposure time control).

5.3 Noise Estimation

We first investigate the initial noise level for the uncorrupted datasets. The synthetic images of Sim do not include noise, while the ground truth (GT) values for KITTI and Udacity are unknown. For an assessment, we apply all three estimators to both datasets and to Sim for comparison.

Analyzing Fig. 5.10, the reference methods estimate the lowest noise level in the Udacity data with a median of $\tilde{\sigma} \lessapprox 1.0$. For KITTI data, B+F provides the largest estimates ($\hat{\sigma} \leq 5\,\mathrm{DN}$), followed by PCA ($\hat{\sigma} \leq 3\,\mathrm{DN}$) and CNN ($\hat{\sigma} \approx 0.75\,\mathrm{DN}$). Nevertheless, all median values are close to $\tilde{\sigma} \approx 1.0\,\mathrm{DN}$. In the case of Sim, B+F and PCA have been distracted by the high density of detailed textures in the images and estimate too high noise levels ($\hat{\sigma} \leq 10\,\mathrm{DN}$ and $\hat{\sigma} \leq 5\,\mathrm{DN}$). This is a well-known phenomenon in traditional noise estimation (cf. Sec. 2.2.2.1) and holds especially for B+F that relies on homogeneous image patches. Only CNN estimates $\hat{\sigma} \leq 1.0\,\mathrm{DN}$ and thus produces the lowest error. Note from all graphs that estimation outliers can be expected, which makes a median estimator more robust than, for instance, a mean estimator. As an intermediate result, the estimations indicate that Udacity and KITTI images contain noise with a median noise level of $\tilde{\sigma} \approx 1.0\,\mathrm{DN}$, which makes both datasets suitable for a median noise estimation evaluation. Moreover, B+F and PCA produce large errors for Sim as these approaches are distracted by texture.

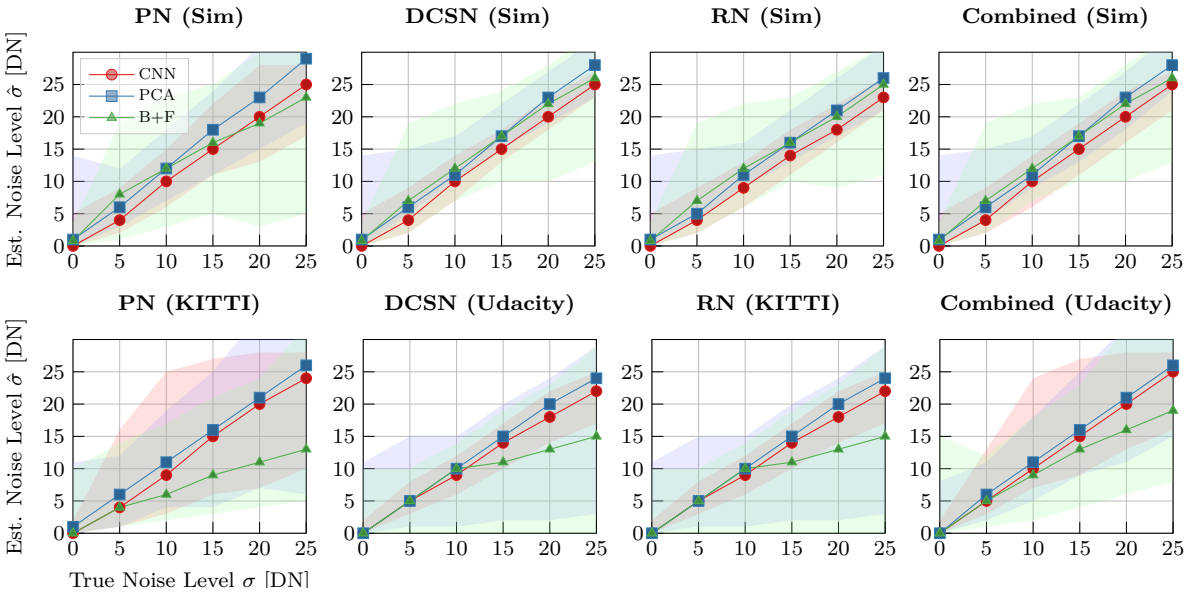


Figure 5.11: *Noise estimation of synthetically corrupted datasets.* Median, minimum and maximum statistics (depicted by sampled points with interpolation in between and the shaded areas, respectively) of the three proposed noise estimators (CNN, PCA, B+F) as the noise level σ increases (from 0 to 25 grayscale levels, DN), for several types of noise (Photon Shot, DCSN, etc.) and datasets (*Sim*, KITTI, Udacity). The last plot shows the effect of combining all noise types (on the Udacity dataset).

Next, we corrupt the datasets with the proposed noise processes and evaluate the three noise estimators by comparing their median, minimum and maximum statistics (rejecting $\approx 5\,\%$ outliers) against the controlled ground truth noise levels. Results are reported in Fig. 5.11. Since we obtained comparable results for KITTI and Udacity, we dropped similar plots.

We first observe from Fig. 5.11 that B+F and PCA methods are prone to structural misestimation: both over-estimate low noise levels, and B+F under-estimates high noise levels. These phenomena have been already reported and are characteristic of the corresponding model family [Shi+05; CZA15]. Moreover, all methods tend to strongly under-estimate noise in natural images, which even reduces the median performance of the B+F method. We observed this behavior in over-exposed areas where most pixels are in saturation, which is expected from vehicle camera images containing large sky areas. The CNN method is less vulnerable since it learned employing fewer meaningful pixels; [Shi+19] omits such image regions under the assumption that under-/over-saturated patches "cannot contain noise" (which only holds for *completely* saturated regions). When comparing results of synthetic and natural images, B+F and PCA produce higher median and minimum noise level estimates for Sim images. This is consistent with the observation from Fig. 5.10 that the high amount of detailed texture in Sim images distracts both estimators and may indicate a too high minimum noise level.

Another observation is the striking difference between the signal-dependent and signal-independent noise cases. Signal-dependent photon shot noise increases the variance of all estimators, especially on real-world data. We observed that large variations in bright and dark intensity areas within one image patch led to over- and under-estimation, respectively. The CNN noise level is limited here since it was trained with $\sigma \leq 30\,\mathrm{DN}$. If all noise types occur simultaneously (combined noise plots in Fig. 5.11), the estimations become more accurate and more robust than in the case of all noise being attributed to photon shot noise. According to the observations of [Xu+18; AB18], realistic noise follows a combined Poisson-Gaussian distribution, and the Poisson part is troublesome for the noise estimators (in particular for those with Gaussian assumptions). Hence, we consider isolated photon shot noise as the worst case scenario. The CNN and PCA methods perform similarly if signal-dependent photon shot noise is included, and the CNN is more reliable (smaller variance) otherwise. In terms of denoising, similar results have been shown by comparing traditional and learning-based methods on real data [Xu+18].

Computational performance

Analogous to Sec. 5.2.1, we perform an intermediate runtime measurement at this point to assess the real-time capability of the noise estimators (more details in Ch. 7). We determined the following average runtimes per image patch: 0.005 s (B+F), 0.002 s (PCA), and 0.002 s (CNN). CNN and PCA executed the fastest, but in the same order of magnitude as the B+F. All noise estimators are real-time capable and considerably faster than blur estimators.

Summary

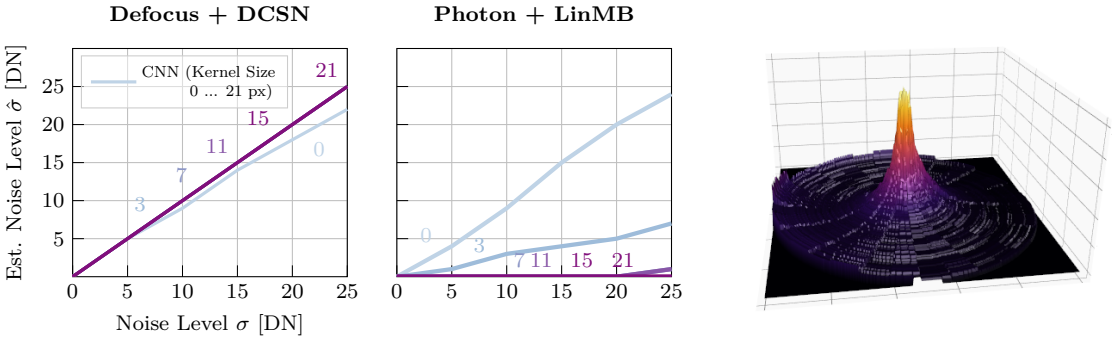
The CNN and PCA noise estimation methods are accurate in median but their reliability decreases the stronger the photon shot noise is. In case of signal-independent noise only, the CNN performs by far most reliably. Since PCA is prone to structural misestimation (e.g., over-exposed areas, small noise levels), we suggest using the CNN for condition monitoring applications. Finally, the reliability of all estimators could be improved by using the median estimation from consecutive frames (Sec. 5.6).

5.4 Estimation of Combined Blur and Noise

Because previous sections showed that CNN blur and noise estimators performed among the best ones on isolated blur/noise cases, we now use these estimators on combined blur and noise corruption experiments. We investigate the cases of combined defocus blur and DCSN (“Defocus + DCSN”), and Photon Shot Noise with simultaneous linear motion blur (“Photon + Lin. Motion”), both on Udacity as the most realistic transportation scenario among our datasets. In both cases, we estimate blur and noise separately.

		Defocus + DCSN					Photon + LinMB				
Size [px]		3	7	11	15	21	3	7	11	15	21
Kernel											
Noise Level	0	2.7	0.9	0.6	0.3	1.4	16.2	10.8	9.8	11.4	7.9
	5	6.0	20.3	10.3	34.5	5.5	3.2	10.5	13.6	14.8	6.3
	10	20.5	49.6	60.4	69.1	70.9	3.1	13.7	16.8	20.6	5.7
	15	23.7	50.2	62.4	69.9	76.5	3.1	15.5	22.4	24.7	14.0
	20	24.0	50.7	62.7	70.6	76.7	3.1	21.3	27.8	25.9	12.1
	25	24.1	50.8	62.5	70.1	76.9	3.1	23.8	31.2	28.1	19.3

Table 5.2: *Blur estimation in the presence of noise* for two image corruption configurations: Defocus + DCSN and Photon + Linear Motion Blur, both on the Udacity dataset. The table contains median blur estimation (AMAE (5.1) in %) for different noise levels and kernel sizes.



(a) Noise estimation results **(b)** PSF reconstructed from MTF

Figure 5.12: *Noise estimation in the presence of blur* for two image corruption configurations: Defocus + DCSN and Photon + Linear Motion Blur, both on the Udacity dataset. (a): Plots of the median noise estimations for different noise levels and blur kernel sizes. Noise estimated for different blur kernel sizes is color-coded from blue to purple. However, differences are almost indistinguishable at this scale. (b): Exemplary PSF reconstruction of a blur estimation (MTF) in the presence of noise that approaches the delta function (assuming Gaussian shape and rotational symmetry).

Blur estimation results are summarized in Tab. 5.2 and those of noise estimation in Fig. 5.12a.

Defocus + DCSN

According to the physics behind the image formation process in Fig. 3.7, an image is corrupted by defocus first and DCSN afterwards. Hence, high-frequency image content is filtered and fully represented by the DCSN. In theory, the larger the blur the easier the noise estimation. This is what we observe in the first plot of Fig. 5.12a. Although there is a small estimation error for zero defocus, $\tilde{\sigma}$ becomes most accurate for $d \geq 3\,\mathrm{px}$ and remains unchanged. Hence, defocus is favorable for DCSN estimation. We expect the same effect for similar combinations of defocus/motion blur and DCSN/readout noise.

On the other hand, DCSN negatively affects defocus estimation because advantageous information for detecting blur (the absence of high frequencies) gets corrupted by noise. We notice two effects from the results of Tab. 5.2: All defocus estimations worsen with increasing noise levels, and this impact becomes more severe with increasing kernel size. While estimations for the smallest and largest kernels ($d \in \{3, 21\}$ px) can be considered as still good for $\sigma = 5$ DN, the same noise level otherwise already leads to poor blur estimations. This outcome was investigated in the context of motion deblurring [TL12], where it was found that, as σ grows, blur estimations approach the Dirac delta function in a large variety of approaches. We observe the same behavior for the CNN estimations, hence the increasing relative error towards larger kernels. Figure 5.12b illustrates an exemplary Dirac delta function reconstructed from an MTF estimation for the configuration $(d, \sigma) = (21\,\text{px}, 25\,\text{DN})$. Generally, defocus estimations are not robust in presence of subsequent noise. Since sensor noise can be detected accurately in case of defocus, a small $\hat{\sigma}$ should be assured before trusting blur estimations.

Photon + Lin. Motion

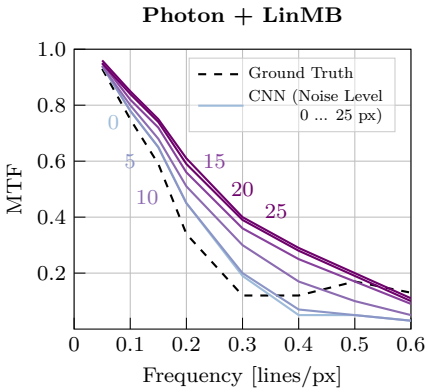
In this case, noise is added before the blur (due to the physics behind the image formation model in Fig. 3.7). Therefore, we expect the opposite behavior, i.e., a poor noise estimation (the blur kernel acts as a classical noise filter) and a good blur estimation. However, only the noise estimation meets the expectations (see the second plot in Fig. 5.12a and Tab. 5.2). A motion blur of size $d = 3$ px already majorly disturbs noise estimation (note that noise is not removed from the image but spread among neighboring pixels). On the other hand, the motion blur leads to structured directional noise (i.e., false image details), which in turn reduces the estimated blur level by increasing $\widetilde{\text{MTF}}$ (Fig. 5.13). This effect intensifies with increasing noise level. Depending on whether blur is overestimated (e.g., for $d = 3$ px) or underestimated (e.g., for $d = 11$ px) when $\sigma = 0$, the AMAE score decreases or increases for higher noise levels, respectively.

We do not observe the same behavior when we replace motion blur with defocus blur ("Photon + Defocus", Fig. 5.14), as defocus blur distributes the noise evenly to the neighboring pixels. The noise still influences the blur estimation of the defocus kernel $d = 3$ px, however, the effect becomes negligible for larger defocus kernels ($d \geq 7$ px).

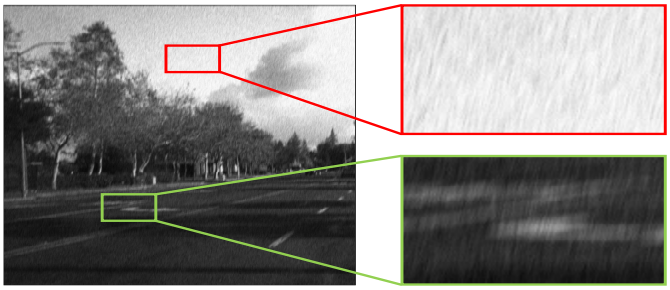
We build upon this finding and propose a simple approach to suppress high noise in order to re-enable the detection of preceding blur. Specifically, we apply an additional defocus filter to estimate preceding small or medium blur on the example of high sensor noise levels $\sigma \geq 10$ DN (Sec. 5.5).

Summary

In summary, we conclude that even a small amount of blur boosts the detection of



(a) Motion est. in presence of PN

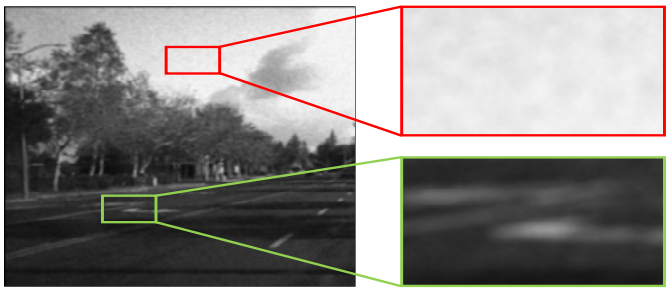


(b) Motion blur induces structured noise

Figure 5.13: *Linear motion blur estimation in presence of preceding photon shot noise* (PN). (a): Increasing noise levels σ increase the MTF estimation and thus decrease the estimated blur level d. (b): Corresponding exemplary image with $(d, \sigma) = (11\,\text{px}, 25\,\text{DN})$ showing structured noise induced by subsequent motion blur.

		Photon + Defocus				
Size [px]		3	7	11	15	21
Kernel						
Noise Level	0	2.7	0.9	0.6	0.3	1.4
	5	0.6	1.1	0.8	0.4	1.3
	10	0.5	1.1	0.5	0.4	1.3
	15	1.2	2.5	0.6	0.4	1.3
	20	5.4	1.7	0.5	0.3	1.4
	25	5.6	1.8	0.5	0.3	1.4

(a) Defocus est. in presence of PN



(b) Defocus blur distributes noise evenly

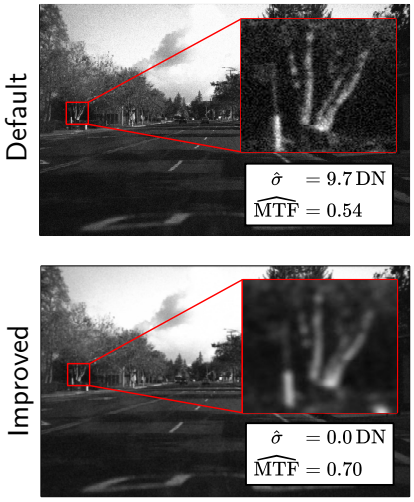
Figure 5.14: *Defocus blur estimation in presence of preceding photon shot noise* (PN). (a): Increasing noise levels σ only significantly increase the AMAE values for the smallest blur kernel $d = 3\,\text{px}$. (b): Corresponding exemplary image with $(d, \sigma) = (11\,\text{px}, 25\,\text{DN})$ showing that subsequent defocus blur distributes the noise evenly to neighboring pixels.

subsequent noise while suppressing preceding noise sources. So, in the presence of blur, photon noise is difficult to estimate and therefore should be avoided. Regarding blur estimation, preceding photon noise can corrupt the result in case of motion blur. Subsequent DCSN with $\sigma \geq 10\,\text{DN}$ already prevents blur estimation, however, it can be re-enabled by applying an additional defocus filter. Hence, if one can minimize photon noise in images[2], we suggest estimating noise before judging a blur estimation result. As in the noise evaluation of Sec. 5.3, sensor noise (DCSN and readout noise) is more favorable than photon shot noise for condition monitoring.

[2]Digitized photon shot noise in images can be mitigated, for instance, by using a camera with a large full-well capacity and by ensuring a well-illuminated scene.

Corruption Levels			Error Metrics			
	d_1 [px]	σ [DN]	MAE (H)	MAE (V)	AMAE	$\text{AMAE}^{\text{Exp.}}$
$d_2 = 7$ px	3	10	2.9	2.0	2.5	16.2
	3	25	3.5	2.1	2.8	16.2
	7	10	12.3	7.7	10.0	10.8
	7	25	11.6	8.8	10.2	10.8
	11	10	11.1	8.4	9.7	9.8
	11	25	14.1	10.1	12.1	9.8
$d_2 = 11$ px	3	10	1.5	0.3	0.9	16.2
	3	25	1.7	0.3	1.0	16.2
	7	10	14.7	12.7	13.7	10.8
	7	25	14.8	12.8	13.8	10.8
	11	10	18.6	14.1	16.4	9.8
	11	25	20.2	15.1	17.6	9.8

(a) Results of improved blur estimation



(b) W/o and w/ improvement

Figure 5.15: *Proposed improved blur estimation in presence of high noise.* (a): Estimation of linear motion blur b_1 (LinMB) on combined pipeline (LinMB + DCSN + Defocus), using Udacity data. The table reports mean absolute errors (MAE) of horizontal (H) and vertical (V) estimations, their average (AMAE), and their expected values ($\text{AMAE}^{\text{Exp.}}$ (5.3)). (b): Exemplary scenario of LinMB + DCSN with $d_1 = 11$ px and $\sigma = 10$ DN. We target estimating the LinMB with a ground truth of $\text{MTF}^{\text{GT}} = 0.75$ (combined image directions at $f = 0.1$). Top of (b): Noise distracts the blur estimation ($\widehat{\text{MTF}} = 0.54$). Bottom of (b): Defocus filtering the noise with $d_2 = 7$ px assists the blur estimation ($\widehat{\text{MTF}} = 0.70$, the influence of defocus was canceled out during the estimation).

5.5 Improved Blur Estimation in Presence of High Noise

The Sec. 5.4 has pointed out that blur is not accurately estimated in the case of high subsequent noise (e.g., DCSN, with $\sigma \geq 10$ DN). Here we demonstrate a simple approach to improve the accuracy of such MTF estimates (Fig. 5.15). The approach exploits that preceding photon noise is not expected to significantly influence the MTF estimation of subsequent defocus blur (see Sec. 5.4). Hence, the approach consists of considering the above-mentioned "high subsequent noise" as the preceding noise of a new blur stage, estimating the overall MTF and reassigning the credit between the two blur stages. Specifically, following up on the Defocus + DCSN case in Sec. 5.4, the considered pipeline has now three stages: LinMB + DCSN + defocus filtering. Letting the first blur kernel be b_1, we filter noise by an additional kernel b_2, estimate the overall blur $\widehat{\text{MTF}}(b_1, b_2) = \widehat{\text{MTF}}(b_1)\,\widehat{\text{MTF}}(b_2)$ and lastly divide the MTF by the known $\text{MTF}^{\text{GT}}(b_2)$ according to the Fourier convolution theorem [Jah00, p. 242]. To this end, we assume $\text{MTF}^{\text{GT}}(b_2) \approx \widehat{\text{MTF}}(b_2)$ and determine the estimation error of $\widehat{\text{MTF}}(b_1)$ with respect to $\text{MTF}^{\text{GT}}(b_1)$.

Due to the combinatorial complexity of the experimental configuration, we focus on the following one grounded on the results from Sec. 5.4: We employ only the CNN

method for MTF estimation on Udacity data, and we consider the case of LinMB + DCSN (representative for sensor noise, to keep it clear and concise). The operating points for this experiment rely on three reasons: (*i*) The choice of b_2's size (d_2) is a trade-off between filtering the noise to reduce its influence on blur estimation without loosing image details necessary to determine b_1. Hence, we pick the smallest defocus filters $d_2 \in \{7, 11\}$ px that lead to stable blur estimation (cf. Fig. 5.14). (*ii*) We consider small/medium motion blur $d_1 \in \{3, 7, 11\}$ px so that the overall blur is still detectable by the CNN. (*iii*) We focus on severe high/higher noise levels $\sigma \in \{10, 25\}$ DN. We next evaluate $\widehat{\mathrm{MTF}}(b_1) \approx \widehat{\mathrm{MTF}}(b_1, b_2)$ / $\mathrm{MTF}^{\mathrm{GT}}(b_2)$ with Fig. 5.15a.

We need to ensure three preconditions to divide $\mathrm{MTF}^{\mathrm{GT}}(b_2)$ from $\widehat{\mathrm{MTF}}(b_1, b_2)$ for a meaningful result: (*i*) $\widehat{\mathrm{MTF}}(b_1, b_2) \leq \mathrm{MTF}^{\mathrm{GT}}(b_2)$, (*ii*) $\mathrm{MTF}^{\mathrm{GT}}(b_2) > 0 + \epsilon$ and (*iii*) $\widehat{\mathrm{MTF}}(b_1, b_2) > 0 + \epsilon$, for all sampled frequencies. We chose the control parameter $\epsilon = 0.05$ to avoid large quotients for small values, and omit frequencies that do not satisfy the conditions.

Figure 5.15a presents results in terms of MAE and AMAE scores (5.1), and their expected values

$$\mathrm{AMAE}^{\mathrm{Exp.}} \doteq \sqrt{\mathrm{AMAE}(\widehat{\mathrm{MTF}}(b_1))^2 + \mathrm{AMAE}(\widehat{\mathrm{MTF}}(b_2))^2} \tag{5.3}$$

from the error propagation of $\widehat{\mathrm{MTF}}(b_1)$ and $\widehat{\mathrm{MTF}}(b_2)$ (cf. Tab. 5.1).

We observe generally slightly worse MAE scores in horizontal than in vertical image direction, which are in agreement with the already-mentioned slight motion blur in the moving direction on Udacity data (the moving direction is closer to the horizontal image axis; see Sec. 5.2.1). It can also be seen that the higher the considered noise and blur levels, the worse the estimations of b_1. The impact of higher noise, which relativizes with increasing d_1, corresponds to the results of Fig. 5.14. Higher blur levels d_1 or d_2 increase the loss of information (where the MTF estimates unrealistically drop below zero) and thus worsen estimations of b_1. This is also why the smaller defocus $d_2 = 7$ px performs better (with results closer to their expected values) and smaller motion blurs d_1 are estimated more accurately (despite their higher expected values). Moreover, the information loss causes the CNN to generally overestimate d_1, which in turn limits the estimation error for $d_1 = 3$ px as its MTF values for the considered frequencies are already close to one. All in all, a defocus filter with $d_2 = 7$ px has been shown to be the best working solution to restore a blur estimation of d_1 in presence of high noise.

Summary

Additional defocus filtering suppresses noise so that estimation of preceding small or medium blur can be re-enabled for high sensor noise levels $\sigma \geq 10$ DN.

This procedure is also suitable for a condition monitoring application as it can be applied in the background without changing the camera configuration.

5.6 Improved Noise Estimation Uncertainty by Temporal Result Aggregation

In previous sections, we identified scenarios in which the blur and noise estimators produce poor estimates (e.g., over-exposure or high texture images, cf. Secs. 5.2.2, 5.2.3 and 5.3). Such estimation errors increase an estimator's uncertainty and thus decrease its trustworthiness, which is undesirable for a robust condition monitoring.

Estimator uncertainty can be reduced in post-processing with low computational overhead by a spatial and/or temporal aggregation of multiple estimations if the underlying blur/noise corruption process is identical over the analyzed spatial domain and/or time span. This post-processing is motivated by spatio-temporal video noise filtering, which aims to increase a video's signal-to-noise ratio [Bra+95].We demonstrate this post-processing on the example of temporal aggregation of median noise estimates generated with a GT noise level $\sigma = 25\,\mathrm{DN}$ for three reasons: (*i*) The focus of this thesis lies on spatially varying corruptions, which favor temporal aggregation (cf. Sec. 1.1). (*ii*) The faster runtimes of the considered noise estimators (compared to the blur estimators) allow for the processing of large datasets, which is favorable for a statistical analysis of potentially large aggregation windows. To keep the combinatorial complexity of the experiment's configuration low, we focus on combined noise (all noise types occur simultaneously) with $\sigma = 25\,\mathrm{DN}$. (*iii*) Median aggregation is robust to outliers within the temporal aggregation window.

Specifically, we apply patch-wise median aggregation to the already calculated patch-wise noise estimations $\hat{\sigma}(i,j)$ (from Sec. 5.3) using aggregation windows of size $n \in \{1,2,4,10,20,40,100\} \doteq A_n$ consecutive frames:

$$\tilde{\sigma}_n(i,j) \doteq \mathrm{median}\left(\hat{\sigma}^{(1)}(i,j),\ldots,\hat{\sigma}^{(n)}(i,j)\right), \tag{5.4}$$

where $\tilde{\sigma}_n(i,j)$ denotes the temporally median aggregated noise estimation of an image patch with index (i,j) and $\hat{\sigma}^{(k)}$ the k-th consecutive image frame. These aggregated median noise estimations are then combined into a probability distribution $P_{\tilde{\sigma}_n}\colon \mathbb{R}_{\geq 0} \to [0,1]$ per dataset, noise estimation method, and aggregation window size ($P_{\tilde{\sigma}_n}$ is assumed to be continuous for simplicity). Following, we analyze the dispersion of each $P_{\tilde{\sigma}_n}$ with increasing aggregation window size n as estimator uncertainty and employ the standard

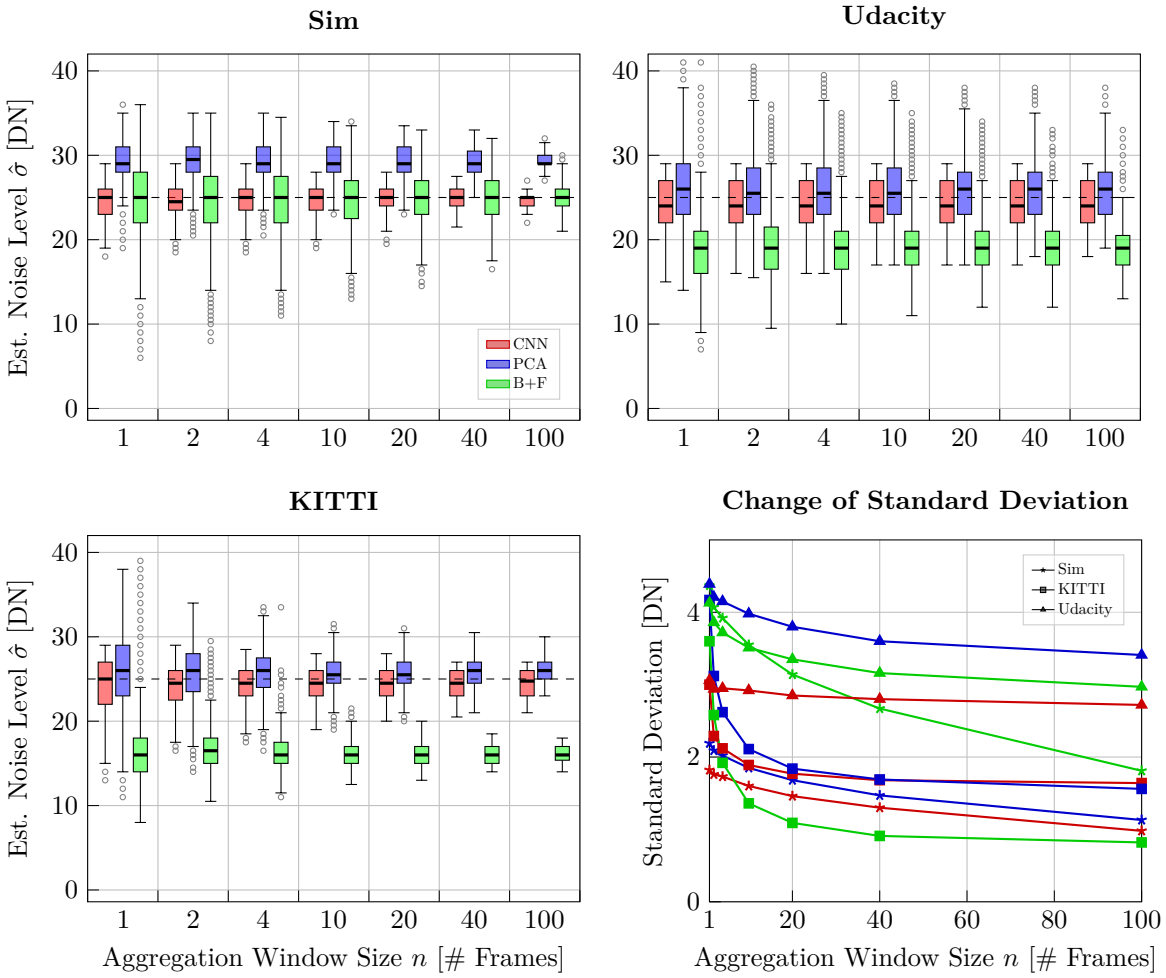


Figure 5.16: *Reduction of noise estimation uncertainty by temporal result aggregation* for a fixed GT noise level of $\sigma = 25\,\mathrm{DN}$ and different numbers of aggregated median estimations for consecutive frames (aggregation windows size). First three plots: Results for all noise estimators and datasets in form of boxplots (bold line: median, dashed line: ground truth, colored area: interquartile range (IQR) between first (Q1) and third (Q3) quartiles, lower whisker: last datum less than $Q1 + 1.5 \times \mathrm{IQR}$, upper whisker: first datum greater than $Q1 - 1.5 \times \mathrm{IQR}$, circle: outlier). Fourth plot: Change in standard deviations of noise estimation distributions as the size of the aggregation window n increases.

deviation $\sigma_{\tilde{\sigma}_n}$ of $P_{\tilde{\sigma}_n}$ as metric for this[3]. Further, we define

$$\begin{aligned} \Delta_{\tilde{\sigma}_{a\to b}} &\doteq \Delta_{\tilde{\sigma}_{1\to a}} - \Delta_{\tilde{\sigma}_{1\to b}}, \\ \Delta_{\tilde{\sigma}_{1\to c}} &\doteq \sigma_{\tilde{\sigma}_1} - \sigma_{\tilde{\sigma}_c} \end{aligned} \tag{5.5}$$

to describe the change of a standard deviation between two aggregation window sizes, with $a, b, c \in A_n \backslash \{1\}$ and $a < b$.

[3]Not to be confused with the "standard error of the median" metric, which quantifies the deviation between medians of a population dataset and a sample, similar to the "standard error of the mean" [LIL15].

The first three plots in Fig. 5.16 depict the temporal aggregation results in form of box plots to put emphasis on the distribution dispersions. One can already see visually that the distributions become narrower and that the spread of outliers generally decreases with increasing window size n. Considering the median values (bold lines), we find that they do not change significantly with increasing aggregation windows size n. This is because the sample median approaches the population median for large sample sizes [Mar18], which is the case for all $n \in [1, 100]$ frames (cf. dataset sizes in Sec. 5.1.1).

The last plot in Fig. 5.16 and Tab. 5.3 detail the changes of the distribution standard deviations. We make two major observations: (*i*) In Sim, B+F benefits the most from increasing window sizes n ($\Delta_{\tilde{\sigma}_{1\to100}} = 2.56\,\mathrm{DN}$), compared to CNN ($\Delta_{\tilde{\sigma}_{1\to100}} = 0.84\,\mathrm{DN}$) and PCA ($\Delta_{\tilde{\sigma}_{1\to100}} = 1.06\,\mathrm{DN}$). This is due to the aforementioned issue of B+F with the texture-rich images in Sim (cf. Sec. 5.3) – the more images of Sim are analyzed, the more likely there are small homogeneous regions from which B+F benefits. B+F profits the most in relative terms, being the most uncertain at $n = 1$ compared to the similarly influenced PCA. (*ii*) In KITTI and Udacity, the initial noise estimation distributions $P_{\tilde{\sigma}_1}$ are comparable, but the corresponding changes of $\sigma_{\tilde{\sigma}_n}$ differ for $n \to 100$. In Udacity, all methods only gain small estimation certainty for $n \to 100$ (CNN: $\Delta_{\tilde{\sigma}_{1\to100}} = 0.34\,\mathrm{DN}$, PCA: $\Delta_{\tilde{\sigma}_{1\to100}} = 0.98\,\mathrm{DN}$, B+F: $\Delta_{\tilde{\sigma}_{1\to100}} = 1.16\,\mathrm{DN}$). In KITTI, however, especially PCA and B+F benefit the most from $n \to 20$ (CNN: $\Delta_{\tilde{\sigma}_{1\to20}} = 1.23\,\mathrm{DN}$, PCA: $\Delta_{\tilde{\sigma}_{1\to20}} = 2.33\,\mathrm{DN}$, B+F: $\Delta_{\tilde{\sigma}_{1\to20}} = 2.51\,\mathrm{DN}$), while neither method improve significantly for $n > 20$ (CNN: $\Delta_{\tilde{\sigma}_{20\to100}} = 0.13\,\mathrm{DN}$, PCA: $\Delta_{\tilde{\sigma}_{20\to100}} = 0.28\,\mathrm{DN}$, B+F: $\Delta_{\tilde{\sigma}_{20\to100}} = 0.27\,\mathrm{DN}$). Among the datasets, we observe the greatest effect of temporal aggregation for KITTI, followed by Sim and Udacity (where Sim provides a larger relative change than Udacity). We attribute this to the scene variation between the consecutive frames. While Sim and Udacity images are recorded contiguously in time and we note comparably higher scene variations between the images of Sim than in Udacity (which is influenced by motion in the scene, camera movement, and exposure time), consecutive KITTI images are independent from each other. Both results support the natural assumption that the more diverse the image content in the analyzed images of the aggregated estimates, the smaller the estimation uncertainties.

Summary

We conclude that temporal result aggregation can reduce estimation uncertainty in online condition monitoring. The results show that, as long as a static noise process can be assumed among the considered frames, the aggregation benefit increases the more the image content varies between the aggregated frames, the larger the aggregation windows size, and the more susceptible the estimators are to specific image content (e.g., texture). We expect a similar benefit for blur estimation.

			Windows Size n for $\Delta_{\tilde{\sigma}_{1\to n}}$ [# Frames]					
Dataset/Method		($\sigma_{n=1}$)	2	4	10	20	40	100
Sim	CNN	1.82	0.06	0.09	0.22	0.38	0.52	0.84
	PCA	2.19	0.10	0.17	0.34	0.51	0.72	1.06
	B+F	4.37	0.30	0.45	0.82	1.23	1.7	2.56
KITTI	CNN	3.00	0.71	0.88	1.11	1.23	1.32	1.36
	PCA	4.17	1.05	1.55	2.06	2.33	2.48	2.61
	B+F	3.60	1.02	1.68	2.24	2.51	2.69	2.78
Udacity	CNN	3.06	0.12	0.11	0.14	0.21	0.26	0.34
	PCA	4.39	0.18	0.24	0.41	0.59	0.79	0.98
	B+F	4.13	0.27	0.41	0.62	0.78	0.97	1.16

Table 5.3: *Change of standard deviations* $\Delta_{\tilde{\sigma}_{1\to n}}$ (5.5) *for increasing aggregation window size* n (with the initial standard deviations $\sigma_{n=1}$). Compare to last plot in Fig. 5.16.

5.7 Discussion

This discussion addresses limitations of our experiments, and provides further considerations on methods and results.

Dataset Design

We suggest that follow-up studies focus on the following five aspects to improve the design of evaluation datasets compared to ours: (*i*) The choice between clean synthetic datasets and initially corrupted data is a trade-off between increasing corruption accuracy and data realism. The latter problem is referred to as the simulation-to-reality gap [Rew+20]. However, there are other image simulators for mobile machines that demonstrate improved data realism, which can be extended by the considered blur and noise corruption processes close to physics [Dos+17; Sha+18]. (*ii*) We reject $\approx 5\%$ outliers in synthetically corrupted datasets to reduce initial corruptions and further consider median statistics for all datasets to handle remaining outliers. Compared to the applied outlier rejection, manual image selection could additionally improve dataset quality. (*iii*) The data for blur estimation is limited due to the high manual labeling effort and the long runtime of the blur estimators, so results could be biased (especially for real-world blur). In addition, more motion blur kernels can be considered to account for motion length and direction. However, this again refers to the trade-off addressed in (*i*). Note that there are also related studies that examine more artificial kernels, such as ones with perfect straight paths [SJA08; Sun+15]. To provide more validity to our results, future work should focus on creating more comprehensive, large-scale blur estimation datasets. (*iv*) Future datasets could also put emphasis to cover more isolated and combined blur and noise cases, which we limited to reduce the combinatorial complexity of the experiments. (*v*) The balance of the dataset in terms of the scenes studied, including image properties and objects of interest, could be considered as well.

General Blur Estimation

On top of the aforementioned initial corruptions of the real-world image datasets, we identified more image content (such as windshield reflections, over-/underexposure, and the availability of image gradients) that also affect blur estimation and thus have to be investigated for a more reliable condition monitoring. Particularly local over- and underexposed areas can adversely influence classic camera exposure controls in that the control tries to alter global camera parameters, which is another motivation for a custom exposure control. A workaround is addressed in Ch. 8. We observed similar critical image content for noise estimation.

This undesired image content is typical for mobile systems operating under natural lighting conditions and can lead to outliers in blur estimation, which we mitigate by using median statistics. In line with this, we evaluate MTF estimates in terms of the (A)MAE score (5.1) that is more robust to outliers than, e.g., the widespread (root-) mean-squared error metric. Moreover, the median value is the optimal minimizer of the MAE metric [Str11, p. 43].

When it comes to the GBB and PMP blur estimators, there are two major improvements necessary before they could be considered for online blur estimation. First, speedups need to be investigated to archive real-time capability, for instance, vectorization or GPU processing. This could enable the usage of GBB/PMP as alternatives to CNN when non-linear motion is expected (e.g., on uneven driving ground at dark scenes that require high exposure times). Second, for accurate blur estimation, the sensitivity of GBB/PMP to image input size, available image content, expected blur kernel size, and kernel complexity must be investigated. Therefore, our results can serve as a basis.

Real-World Blur Estimation

We propose two datasets DEFCARS and MOTCARS for real-world blur estimation and determined ground truth values manually on a reconstructed PSF (Fig. 5.3). This PSF relies on the assumptions of rotational symmetry and a Gaussian shape to compensate for the missing phase transfer function (3.20). However, the symmetry assumption may be too simplistic for real-world PSFs of cameras [Dub+17]. Moreover, the determination of reproducible blur kernel sizes is more difficult for a Gaussian shape than for a uniform kernel shape, which we assume in (3.6). As a criterion, we require the kernel sizes to include $\approx 95.4\%$ of the Gaussian, i.e., four standard deviations, which we could determine using the resolving power tool labels.

In contrast to DEFCARS, MOTCARS includes significantly fewer image patches (75 compared to 4k) due to the lower spatial resolution of the used camera and the higher GT labeling overhead.

The first shortcoming is rooted in the fact that we installed a camera with a programmable API to trigger automatic image acquisition and camera parameter changes on the robotic platform (especially in preparation for Ch. 7). The second reason is the evaluation of two SLEs per Siemens star. We also like to point out that SLE angles marginally differ between the images due to the slightly varying camera movements. Since the used Siemens star with 32 segments provides edges in discrete increments of $360°/\,32\,\text{segments} = 11.25°/\text{segment}$, we chose positions closest to the suggested $\theta \in [5, 180 + 5]\,°$ (cf. Sec. 3.3.1.1), but this decreases comparability as the analyzed directions can differ by at most $11.25°/\,2 = 5.625°$. This only affects MOTCARS data with ellipsoid PSF shapes. Also note that for better comparison, only images of MOTCARS containing the same parts of the scene are selected, and that the dataset is mainly limited to horizontal movements.

Improved Blur Estimation in Presence of high Noise

The proposed defocus filter post-processing comes with limitations as well. The filter size needs to be chosen carefully to avoid information loss, since high image details (i.e., high image frequencies) can be irreversibly filtered (when the corresponding MTF approaches zero). Further research could empirically determine adaptive defocus blur sizes for different noise levels and total image blur estimations. Also note that this technique is not applicable to large initial blur. For an online condition monitoring, we suggest to detect and tackle small blurs before they become more severe (if possible), or otherwise to initiate actions outside the camera system (e.g., in case of large defocus blur: request human interaction to inspect the lens system for a possible defect; in case of large motion blur: request spotlight activation and exposure time reduction in dark scenes, or speed adjustment of the platform).

Improved Estimation Uncertainty

The assumption of spatially varying corruptions can be weakened if they can be excluded (for instance, noise can be assumed spatially constant when imaging static scenes with a temperature regulated camera system). Temporal and spatial aggregation window sizes can also compensate each other, which enables to make weaker assumptions of temporal or spatial consistency of the corruptions.

In recent years, researchers also addressed the reduction of estimation uncertainty directly at its source (e.g., to reduce CNN model uncertainty by integrating further techniques of Physics-ML or to reduce data uncertainty by CNN training data augmentation). The interested reader is referred to [Gaw+21].

5.8 Summary

In this chapter, we commenced to evaluate the condition monitoring part of the proposed self-health-maintenance framework, specifically image blur and noise estimation, and addressed individual shortcomings.

We first introduced five datasets in Sec. 5.1: three with synthetic blur/noise corruptions (Sim, KITTI, and Udacity), and two self-recorded with real-world defocus and motion blur, respectively (DEFCARS and MOTCARS).

The subsequent Sec. 5.2 focused on blur estimation. We found that CNN is best suited for online condition monitoring when it comes to real-time requirements, defocus, linear motion, or generally small blurs. In contrast, the traditional estimators (GBB and PMP) operate best for complex non-linear motion in non-real-time scenarios. Results on real-world corrupted data further revealed that CNN decreases in accuracy when initial inherent blur of real camera systems is taken into account. Challenging scenes of under- and overexposure reduced the performance of all estimators. For more robust estimates, it is recommended to employ median statistics over small time spans and the variance to assess estimation uncertainty.

Noise estimation experiments in Sec. 5.3 showed that the learning-based CNN archived the best results in terms of accuracy in all scenarios. However, the accuracy of CNN and the traditional estimators (PCA and B+F) decreased as photon shot noise in the image data increased. We also identified structural mis-estimations for all methods (in cases of overexposure), B+F and PCA (overestimate small noise levels), and B+F (underestimates high noise levels). All noise estimators were considerably faster than the blur estimators, with CNN and PCA being the fastest.

In Sec. 5.4, we investigated blur and noise estimation when both corruptions are present simultaneously, and found that blur and noise affect each other's estimates: (*i*) Blur boosts the detection of subsequent noise but suppresses preceding noise. (*ii*) Preceding noise and high subsequent noise ($\sigma \geq 10\,\mathrm{DN}$) both corrupt blur estimation. As for isolated noise estimation, photon shot noise is less favorable than sensor noise.

Section 5.5 introduced an improved blur estimation in presence of high subsequent noise. We demonstrated that an additional defocus filter can suppress the noise to re-enable estimation of small and medium sized motion blur without changing the camera configuration. For this improved condition monitoring, we concluded to estimate noise before judging blur estimation, if photon shot noise could be minimized.

In Sec. 5.6, we examined the reduction of noise estimation uncertainty by temporal result aggregation. It could be demonstrated that median aggregation of patch-wise estimations for consecutive frames can reduce uncertainty under the assumption of a static corruption process within the considered aggregation window.

The benefit of this improvement increases the better this assumption, the larger the windows size parameter, and the more sensitive an estimator is to specific occasional image content (e.g., high texture density).

The last Sec. 5.7 addressed improvements for the blur estimators (e.g., vectorize GBB and PMP) and the noise estimators (e.g., avoid photon shot noise or improve its estimation). We also raised awareness about limitations of improved blur estimation (too large post-filter sizes) and concluded to target blur when it is still small, or otherwise try to initiate actions outside the camera system (e.g., request for human interaction or active headlights on a vehicle). Lastly, the improvement of estimation uncertainty can be extended to spatial aggregation as well.

CHAPTER 6

Evaluation: Noise Source Estimation

The underlying idea of noise source estimation is to employ camera metadata in addition to the captured image to determine the noise contribution of each noise source from within a camera system in order to find adequate countermeasures (Sec. 4.3.3). Noise source estimation is considered the second part of our proposed condition monitoring module – this chapter covers its evaluation. We would like to emphasize that this chapter also provides real-world noise evaluations that complement the experiments from Sec. 5.3.

We first describe the datasets used and the image noise applied (Sec. 6.1). Depending on whether a dataset includes ground truth (GT) labels or not, we conduct either quantitative or qualitative experiments. Our quantitative experiments comprise performance evaluations on simulated and real-world data (Sec. 6.2). We further demonstrate our methods in real field campaigns and on three use cases of unexpected noise (Sec. 6.3). Besides the ability to quantify individual noise sources, we subsequently demonstrate the improved total noise estimation performance on the downstream task of real-world image denoising (Sec. 6.4). Next, we examine the effects of each of the individual camera metadata on the estimated noise and compare them to the theoretical noise model (Sec. 6.5). Lastly, shortcomings and potential extensions are briefly addressed (Sec. 6.6) and the chapter is summarized (Sec. 6.7). This chapter is partially published in [Wis+23a].

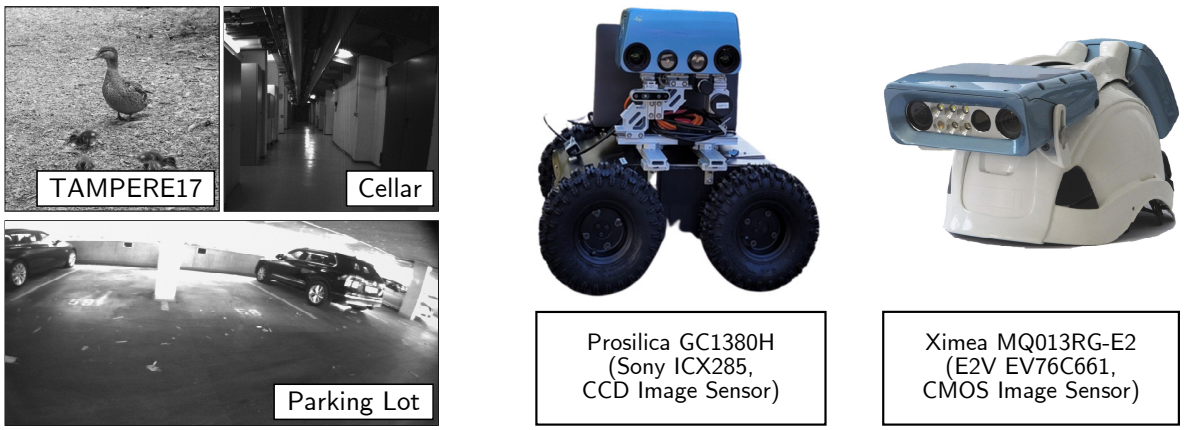
We compare our proposed estimators against:

(*i*) *B+F* [Shi+05], $DRNE_{\text{cust.}}$, *PCA* [CZH15], and *PGE-Net* [BCM21] in the case of σ_{Total},

(*ii*) *PGE-Net* for σ_{PN}, and

(*iii*) noise model [KW14] predictions from the respective metadata for all individual noise levels $\sigma_{i\in\{\text{PN, DCSN, RN}\}}$.

Note that *PGE-Net* is only applicable in the quantitative experiments, since it requires (uncorrupted) GT images in order to calculate estimations $\hat{\sigma}_{i\in\{\text{PN, Total}\}}$.



(a) Datasets **(b)** Camera Systems

Figure 6.1: *Datasets and camera systems.* (a): Exemplary image snippets from new datasets that complement *Sim*, KITTI and Udacity (see Fig. 5.1a), namely TAMPERE17 (512×512 px), *Cellar*, and *Parking Lot*. *Cellar* and *Parking Lot* are both acquired with the camera systems *ICX285* (1360×1024 px) and *EV76C661* (1280×1024 px). (b): *ICX285* is attached on an autonomous robotic platform and *EV76C661* on an inspection helmet.

All experiments are executed on an Intel Xeon W-2145 CPU and an NVIDIA Quadro RTX 6000 GPU, with the neural networks running on the GPU.

6.1 Datasets

We augment four datasets with ground truth labels and two datasets with pseudo ground truth labels (Fig. 6.1a).

6.1.1 Datasets with Ground Truth

We employ one simulated and three real-world datasets: *Sim*, KITTI [GLU12], Udacity [Uda16], and TAMPERE17 [Pon+18]. The uncorrupted images from Sim, KITTI and Udacity are reused as described in Sec. 5.1.1. Similar to our training dataset TAMPERE21 (Sec. 4.3.3), TAMPERE17 provides 300 natural images (4.8k patches of size 128×128px) with a controlled noise level of $\sigma^2 < 1$ DN. From TAMPERE17 we use the grayscale version.

Noise Generation Overview

We corrupt all datasets with *simulated* or *real-world* noise. In the simulated case, we added noise to the images like in our training dataset (cf. Sec. 4.3.3). Note that in contrast to the (total) noise generation proposed in Ch. 5, we do not generate fixed random noise levels but random noise model metadata (i.e., random camera sensors in random states) and omit raw noise amplification to obtain matching pairs of metadata and corresponding noise levels. In the real-world case, we generated in total 12k RN and DCSN image tuples $(I_{\mathrm{RN}}, I_{\mathrm{DCSN}})$ with about 600 different metadata sets from two

different camera systems that we abbreviate according to their implemented camera sensors: *ICX285* [Gmb21] and *EV76C661* [Xim23] (Fig. 6.1b). The first one is considered a scientific-grade CCD and the latter an industrial-grade CMOS camera system. PN is calculated synthetically as the quantum nature of light determines PN to strictly follow the Poisson distribution.

Real-World Noise Generation

The real-world noise generation took place in a darkroom with closed camera apertures to prevent light signal (i.e., we took dark frames). Moreover, we disabled all image post-processing and used the highest possible camera bit depths (12 and 10 bit) to minimize quantization errors. To generate RN, we set the exposure time to the minimum of 0.001 s to counteract dark current integration, applied a random camera gain from $[0, 24]$ dB, and generated multiple image sequences. In order to generate corresponding DCSN images right after an RN image sequence, we sampled another exposure time from $[0.001, 0.2]$ s but kept the same gain (note that RN is still included in the DCSN images at this point).

Real-World Noise Processing

We identify three issues with the raw noise images that needed to be addressed in post-processing: (i) The DCSN and RN intensity distributions may be truncated with all negative intensity values set to zero, which supports our zero-camera-offset assumption, but also affects the ground truth noise level determination. We tackle this issue in four steps: (a) In each noise image histogram we determine the bin $x_{\max}$ that corresponds to the distribution maximum, (b) mirror histogram bins $x \geq 2x_{\max}$ along the vertical axis at $x_{\max}$ to reconstruct bins $x \leq 0$, (c) fit a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ into the fixed histogram, and (d) sample a new noise image from $\mathcal{N}(\mu, \sigma^2)$. (ii) Secondly, the images I_{DCSN} still contain RN. We approach this issue during the step $(i.c)$ by calculating a rectified Gaussian distribution $\mathcal{N}(\mu_{\text{DCSN}^*}, \sigma^2_{\text{DCSN}^*})$ with

$$\begin{aligned} \mu_{\text{DCSN}^*} &\doteq \mu_{\text{DCSN}} - \mu_{\text{RN}}, \\ \sigma^2_{\text{DCSN}^*} &\doteq \sigma^2_{\text{DCSN}} - \sigma^2_{\text{RN}}, \end{aligned} \tag{6.1}$$

following the central limit theorem for the addition of two statistically independent Gaussian distributed random variables [Dev11, pp. 230–232]. Corrected DCSN images in $(i.d)$ are then sampled from this rectified distribution. (iii) Lastly, we observe residual dark current in images I_{DCSN}. To counteract this, for each image set $I_{i \in \{\text{DCSN, RN}\}}$ we calculate pixel-wise means from the respective first 20 images and remove this mean image from the remaining images in the set, which form the final dataset. The whole real-world noise processing is detailed in App. A.3 (Alg. 1).

6.1.2 Datasets without Ground Truth

We collect two datasets from field campaigns without ground truth labels: *Cellar* and *Parking Lot*. Both datasets contain about 1000 grayscale images from respective eponymous environments and were recorded with both camera systems *ICX285* and *EV76C661*. We ensured high noise levels by applying the minimum exposure time of 0.001 s (to capture low but detectable signals), maximum gain of 24 dB (to strongly amplify signal and noise without saturation), and by disabling image post-processing (that could reduce noise).

Experiments on Unexpected Noise

We evaluate the fourth noise type $\xi_{\mathrm{M/I}}$ (Sec. 4.3.3) 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: $\xi_{\mathrm{M/I}} = 0$ and $\xi_{\mathrm{M/I}} \neq 0$. The case $\xi_{\mathrm{M/I}} \neq 0$ is further subdivided into $\xi_{\mathrm{M/I}} < 0$ and $\xi_{\mathrm{M/I}} > 0$. For $\xi_{\mathrm{M/I}} < 0$, we simulate an additional image noise source by adding randomly generated Gaussian noise $\mathcal{N}(\mu = 0, \sigma = 5\,\mathrm{DN})$ to the images. For $\xi_{\mathrm{M/I}} > 0$, we increase the model noise by synthetically doubling the value of the camera metadata *thermal white noise*. This parameter adjustment can 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 demonstrate the case of doubling the metadata *sensor temperature*.

Experiments on Denoising

We also employ *Cellar* and *Parking Lot* to examine the noise source estimators in terms of denoising[1]. Therefore, we select 50 consecutive image frames per camera and dataset that depict a static scene. In order to create pseudo-ground truth images with minimized temporally varying noise, we follow the standard real-world denoising studies [LTO14; ZCH16; Zha+19] and pixel-wisely average each 50 consecutive image frames. Similarly inspired by [Zha+19], different smaller real-world noise levels are generated by averaging $n \in \{1, 2, 4, 8, 16\}$ consecutive image frames per sub-dataset. All in all, we obtain 20 sub-datasets for denoising (two scenes, two cameras, and five noise levels).

[1] Keep in mind that we need the camera metadata and therefore cannot use standard datasets.

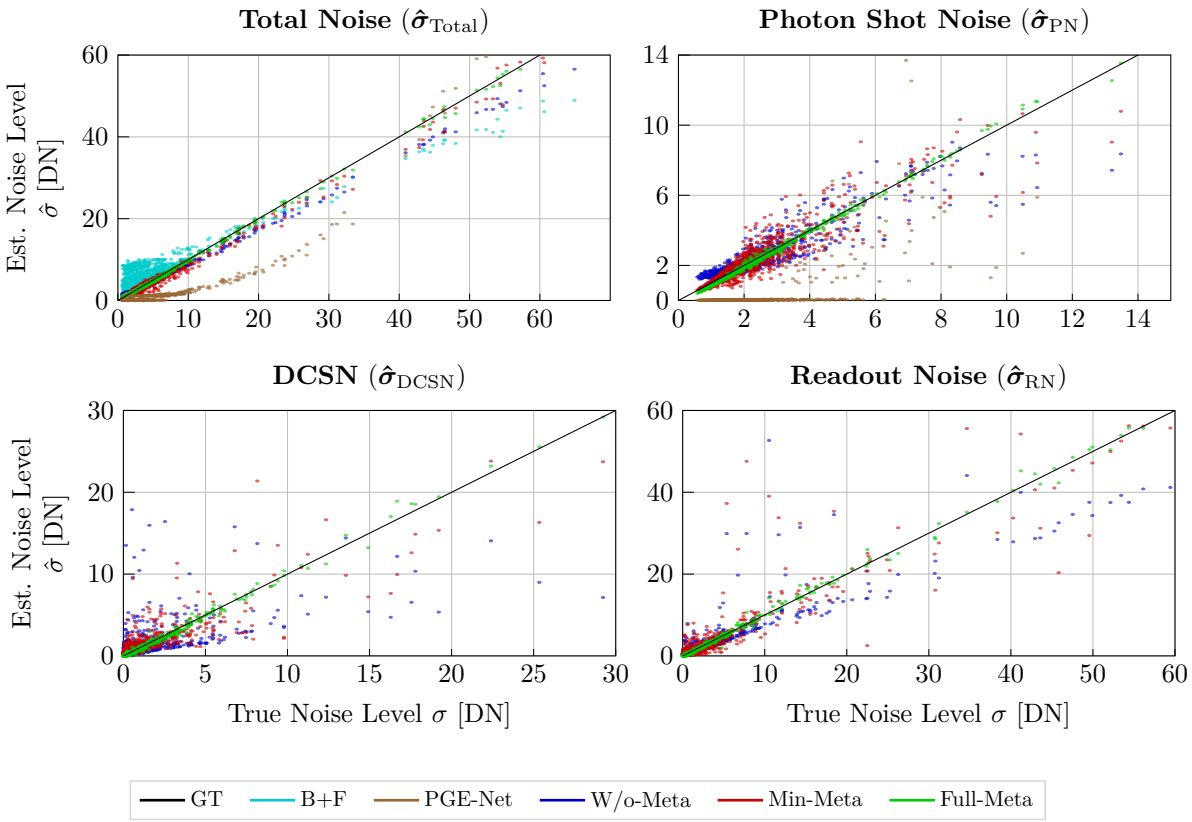


Figure 6.2: *Noise source estimation on synthetic noise (dataset:* Sim*, camera: random).* Each dot represents the mean noise estimation of one image. The plots of $DRNE_{\text{cust.}}$ and *PCA* are omitted in the case of $\hat{\sigma}_{\text{Total}}$ due to a strong similarity with the other plots (to avoid clutter).

6.2 Quantitative Experiments

We follow [CZA15] and evaluate our noise source estimators in terms of accuracy (Bias), robustness (Std), and overall performance (RMS):

$$\begin{aligned} \text{Bias} &\doteq \left|\mathbb{E}[\sigma - \mathbb{E}(\hat{\sigma})]\right|, \\ \text{Std} &\doteq \sqrt{\mathbb{E}[(\hat{\sigma} - \mathbb{E}(\hat{\sigma}))^2]}, \\ \text{RMS} &\doteq \sqrt{\text{Bias}^2(\hat{\sigma}) + \text{Std}^2(\hat{\sigma})}, \end{aligned} \tag{6.2}$$

where $\hat{\sigma}$ is the estimated noise level and σ is the true noise level. Smaller RMS, Bias, and Std values indicate better performance.

6.2.1 Simulated Noise

The performance on the synthetically-added noise datasets is summarized in Tab. 6.1, while mean noise estimation results on *Sim* are depicted in Fig. 6.2.

Let us focus on results from Tab. 6.1 first. Among the reference methods, we observe that *PGE-Net* performs the worst due to underestimation (cf. Fig. 6.2), which agrees with

		Photon Shot Noise			DCSN			Readout Noise			Total Noise		
		Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS
Sim	B+F	-	-	-	-	-	-	-	-	-	2.51	3.00	3.91
	$DRNE_{cust.}$	-	-	-	-	-	-	-	-	-	**0.07**	**0.23**	**0.23**
	PCA	-	-	-	-	-	-	-	-	-	0.75	1.07	1.30
	PGE-Net	1.74	3.02	3.49	-	-	-	-	-	-	3.23	4.36	5.43
	W/o-Meta	**0.01**	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	**0.07**	**0.09**	**0.07**	**0.34**	**0.35**	**0.09**	**0.46**	**0.47**	0.16	0.29	0.33
KITTI	B+F	-	-	-	-	-	-	-	-	-	0.00	2.09	2.09
	$DRNE_{cust.}$	-	-	-	-	-	-	-	-	-	0.18	**0.22**	**0.28**
	PCA	-	-	-	-	-	-	-	-	-	1.74	1.36	2.21
	PGE-Net	2.03	1.16	2.33	-	-	-	-	-	-	4.00	4.80	6.24
	W/o-Meta	0.16	0.67	0.69	0.18	2.36	2.37	0.61	2.33	2.41	0.32	0.98	1.04
	Min-Meta	**0.04**	0.66	0.66	0.14	1.50	1.51	**0.04**	1.92	1.92	**0.01**	0.78	0.78
	Full-Meta	0.11	**0.14**	**0.18**	**0.05**	**0.31**	**0.32**	0.10	**0.38**	**0.40**	0.03	0.34	0.35
TAMPERE17	B+F	-	-	-	-	-	-	-	-	-	2.22	4.19	4.74
	$DRNE_{cust.}$	-	-	-	-	-	-	-	-	-	0.21	0.44	0.49
	PCA	-	-	-	-	-	-	-	-	-	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	**0.07**	3.11	3.11	**0.02**	3.04	3.04	**0.02**	1.18	1.18
	Min-Meta	**0.09**	0.82	0.83	0.21	2.01	2.02	0.39	3.73	3.75	**0.02**	1.05	1.05
	Full-Meta	0.10	**0.13**	**0.16**	0.09	**0.29**	**0.30**	0.17	**0.37**	**0.41**	0.05	**0.43**	**0.43**
Udacity	B+F	-	-	-	-	-	-	-	-	-	1.09	2.19	2.44
	$DRNE_{cust.}$	-	-	-	-	-	-	-	-	-	0.24	0.50	0.54
	PCA	-	-	-	-	-	-	-	-	-	0.70	0.93	1.17
	PGE-Net	1.58	2.05	2.59	-	-	-	-	-	-	3.04	3.70	4.79
	W/o-Meta	**0.05**	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	**0.03**	2.21	2.21	0.27	2.38	2.40	**0.11**	0.88	0.89
	Full-Meta	0.06	**0.14**	**0.15**	0.04	**0.30**	**0.30**	**0.10**	**0.44**	**0.45**	0.14	**0.42**	**0.45**

Table 6.1: *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.

the observation from the original authors [BCM21]. We can further see that $DRNE_{cust.}$ generally produces better results than *PCA* for all metrics, and both yield better results than *B+F*. This observation matches the results from Sec. 5.3. Considering our proposed methods, we observe that all three estimators accurately and robustly determine σ_{Total}, where *Full-Meta* is generally the best, and *Full-Meta* and *w/o-Meta* perform slightly more robust than *Min-Meta* (smaller Std). In comparison to the reference methods, *Full-Meta* is on par with $DRNE_{cust.}$. When it comes to noise source estimation, *Full-Meta* performs the best. Both accuracy and robustness span intensity levels below the 1 DN resolution for all three noise sources in all three datasets. *w/o-Meta* and *Min-Meta* also accurately quantify the single noise types within sub-intensity levels on average (small bias). However, they have 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 make two detailed observations: all three methods estimate PN

the best, and *Min-Meta* determines the DCSN amount more robustly than *w/o-Meta*. We attribute 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* obtains the camera's *full well capacity* parameter, which seems to slightly improve PN estimation. The more robust DCSN estimation performance of *Min-Meta* can be ascribed to its access to *temperature* and *exposure time metadata*, since both have a major impact on thermal noise [KW14]. The significance of metadata on separating the noise sources is further underpinned by the minor performance on RN estimation (as the minimal metadata only have a minor impact on the noise model) and by the prevailing performance of *Full-Meta*, which has access to the largest amount of metadata.

Figure 6.2 confirms the results of Tab. 6.1. It further indicates an increasing bias for *w/o-Meta*, and an increasing Std (spread of the distributions) for *w/o-Meta* and *Min-Meta* with increasing noise levels $\sigma_{i\in\{\text{Total, PN, DCSN, RN}\}}$.

Computational Cost

The computation time is determined by averaging the noise estimation inference times for 13.5k Udacity image patches (i.e., 100 Udacity images). We repeated the measurements 5 times and took the average to mitigate 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 ($DRNE_{\text{cust.}}$), and 0.1 ms (*PGE-Net*). The small differences of the proposed methods are in accordance with their similar number of network parameters. Note that *PGE-Net* is faster because it processes a whole image at once, but it does not estimate as many noise sources nor is as accurate as the proposed method(s).

Summary

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

6.2.2 Real-World Noise

Next we discuss the estimation performances of the real-world DCSN/RN produced by *ICX285* and *EV76C661* using Tab. 6.2 and Fig. 6.3. Note that these two noise-optimized sensors produce 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 focus on *ICX285* here and consider *EV76C661* in App. B.2.3 (Tab. B.5).

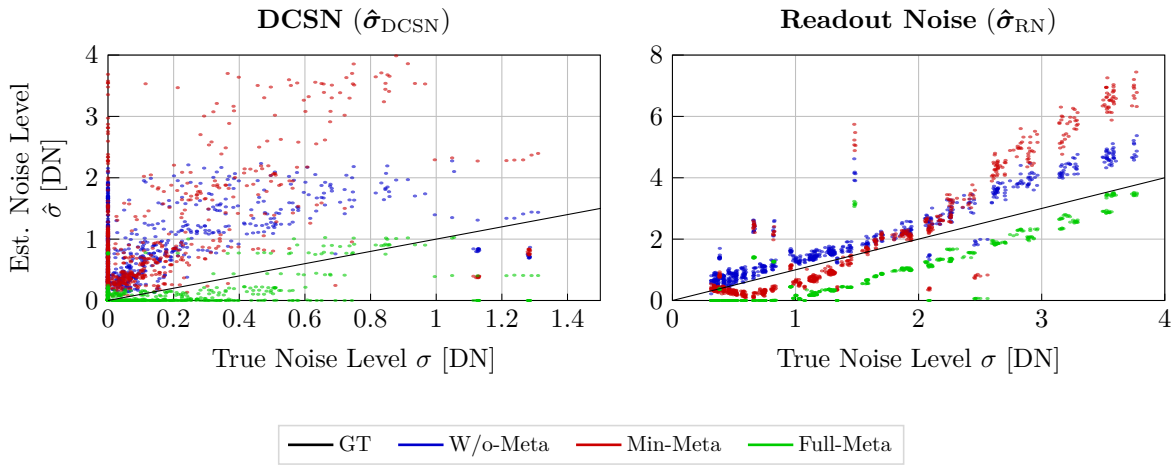


Figure 6.3: *Noise source estimation on real-world noise (dataset:* Sim*, camera:* ICX285*).* Compare to Fig. 6.2.

In contrast to the fully simulated noise experiments (Tab. 6.1), the absolute DCSN and RN estimation performances of *w/o-Meta* and *Min-Meta* seem to have improved in Tab. 6.2. These results should not be overrated due to the generally smaller noise levels and because the errors in the fully simulated cases started to majorly increase for noise levels $\sigma_{i\in\{\mathrm{DCSN,RN}\}} \geq 5\,\mathrm{DN}$. However, we observe two significant relative performance changes: *Full-Meta* worsened for RN, and *w/o-Meta* improved for DCSN/RN. We attribute 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. Fig. 6.3), because both methods are trained on simulated data only where it has been shown that *Full-Meta* matches it better (Tab. 6.1). In the case of DCSN, the better performance of *w/o-Meta* is misleading, since only *Full-Meta* seems to approximately fit the ground truth, while the others fail (see Fig. 6.3). These errors also propagate to the overall noise estimation. The estimations of the simulated PN have not changed significantly.

Summary

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

6.3 Experiments on Real-world Platforms

We recorded datasets *Cellar* and *Parking Lot* with camera systems *ICX285* and *EV76C661* in field campaigns (Fig. 6.1b). For comparison, we use *B+F*, $DRNE_{\mathrm{cust.}}$, 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 focus on *ICX285* and *Cellar* here, and consider the rest in App. B.2.3. We first evaluate the raw dataset (Sec. 6.3.1) and subsequently test three altered versions with unexpected noise (Sec. 6.3.2).

		Photon Shot Noise			DCSN			Readout Noise			Total Noise		
		Bias	Std.	RMS	Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS
Sim	B+F	-	-	-	-	-	-	-	-	-	3.12	1.60	3.51
	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.17	0.28	0.33
	PCA	-	-	-	-	-	-	-	-	-	1.11	0.82	1.38
	PGE-Net	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	**0.61**	**0.75**	0.08	0.27	0.29
	Min-Meta	1.03	0.21	1.05	0.80	0.86	1.18	**0.26**	1.35	1.38	0.77	0.65	1.00
	Full-Meta	**0.14**	**0.09**	**0.17**	**0.15**	**0.45**	**0.47**	0.82	0.95	1.25	**0.04**	**0.19**	**0.20**
KITTI	B+F	-	-	-	-	-	-	-	-	-	0.30	0.64	0.71
	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	**0.09**	0.32	0.33
	PCA	-	-	-	-	-	-	-	-	-	0.53	0.68	0.86
	PGE-Net	2.51	1.02	2.71	-	-	-	-	-	-	3.03	1.50	3.41
	W/o-Meta	0.66	0.57	0.88	0.53	0.60	0.80	0.03	**0.64**	**0.64**	0.10	**0.03**	**0.11**
	Min-Meta	0.83	0.19	0.85	0.69	0.78	1.05	**0.02**	1.27	1.27	0.12	0.31	0.34
	Full-Meta	**0.10**	**0.12**	**0.16**	**0.16**	**0.49**	**0.51**	0.90	1.10	1.43	0.45	0.76	0.89
TAMPERE17	B+F	-	-	-	-	-	-	-	-	-	2.71	3.54	4.43
	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.37	0.40	0.55
	PCA	-	-	-	-	-	-	-	-	-	3.07	2.77	4.14
	PGE-Net	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	**0.76**	**1.06**	0.26	0.53	0.59
	Min-Meta	0.95	0.28	0.99	0.85	0.82	1.18	**0.37**	1.36	1.41	0.59	0.78	0.98
	Full-Meta	**0.22**	**0.14**	**0.26**	**0.14**	**0.41**	**0.44**	0.85	0.87	1.21	**0.13**	**0.36**	**0.38**
Udacity	B+F	-	-	-	-	-	-	-	-	-	0.33	0.58	0.66
	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	**0.01**	0.53	0.53
	PCA	-	-	-	-	-	-	-	-	-	0.14	0.63	0.64
	PGE-Net	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	**0.27**	**0.65**	**0.70**	0.04	**0.27**	**0.27**
	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	**0.04**	**0.10**	**0.11**	**0.17**	**0.44**	**0.47**	0.87	0.97	1.30	0.25	0.30	0.39

Table 6.2: *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.

6.3.1 Expected Noise ($\sigma_{\text{Model}} \approx \sigma_{\text{Image}}$)

Let us first focus on the noise source identification (top row in Fig. 6.4). We see that *Full-Meta* matches the noise model best with $|\hat{\sigma}_{\text{Full-Meta}} - \hat{\sigma}_{\text{Noise Model}}| < 1\,\text{DN}$ in each noise case, followed by *Min-Meta* and *w/o-Meta*. These results are generally in accordance to the simulated noise evaluations in Sec. 6.2.1. The only significant difference we observe is that *Min-Meta* matches the relative value range of the PN noise model curve better than *w/o-Meta* (smaller Std). This can be explained with the *camera gain* parameter that *Min-Meta* obtains as one key parameter in the noise model to determine PN (already indicated on the simulated *ICX285* in Tab. 6.2). The residual noise plot depicts 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* indicates that it has not learned to detect any residual noises. From this residual noise estimation of *Full-Meta* (and later addressed observations that *Full-Meta* is able to quantify residual noise correctly),

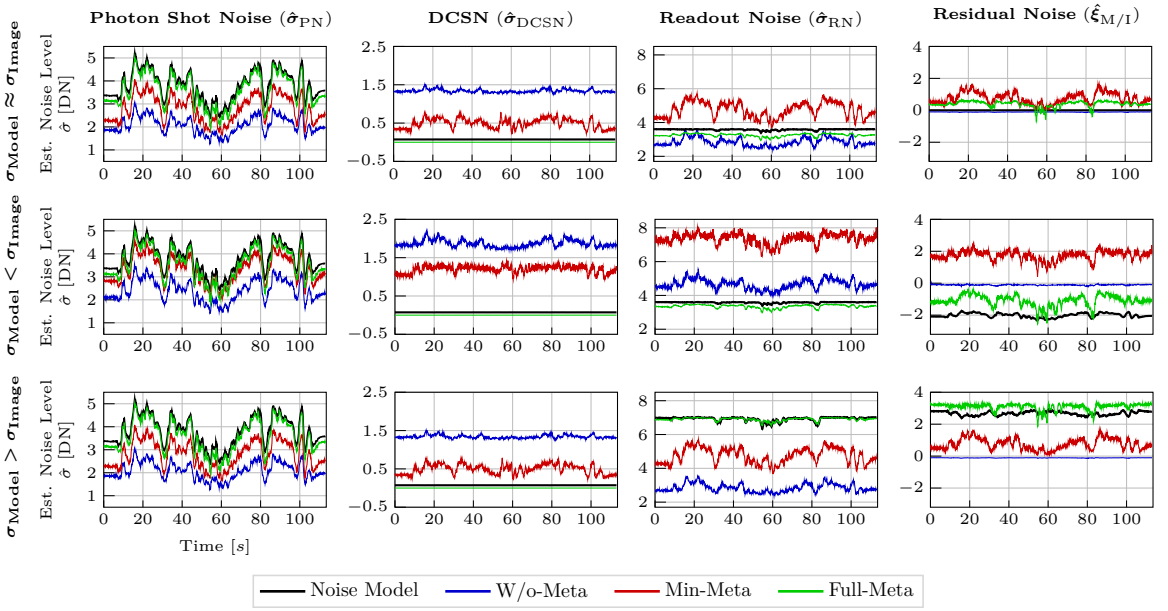


Figure 6.4: *Noise source estimation with and without unexpected noise (dataset:* Cellar*, camera:* ICX285*).* Top row: Estimation on the uncorrupted dataset. Middle row: Image noise increased by random Gaussian noise $\mathcal{N}(\mu = 0, \sigma = 5\,\mathrm{DN})$. Bottom row: Model noise increased by doubling camera parameter *thermal white noise*.

we assume for *Cellar* that

$$\xi_{\mathrm{M/I}} \approx 0 \overset{(4.3)}{\Longrightarrow} \sigma_{\mathrm{Model}} \approx \sigma_{\mathrm{Image}}. \tag{6.3}$$

In the total noise inspection (Fig. 6.5), we consider only *Full-Meta* as the overall best of our proposed methods. We see from both plots that *Full-Meta* produces similar estimations as the reference methods. Hence, we consider its results as plausible.

Summary

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

6.3.2 Unexpected Noise ($\sigma_{\mathbf{Model}} \neq \sigma_{\mathbf{Image}}$)

Here we evaluate three scenarios where we synthetically increase image noise or model noise to reach $\xi_{\mathrm{M/I}} < 0$ (i.e., $\sigma_{\mathrm{Model}} < \sigma_{\mathrm{Image}}$) or $\xi_{\mathrm{M/I}} > 0$ (i.e., $\sigma_{\mathrm{Model}} > \sigma_{\mathrm{Image}}$), respectively. We investigate these scenarios on the basis of the raw *Cellar* dataset for that we assume that the applied noise model follows the actual image noise (i.e., (6.3): $\xi_{\mathrm{M/I}} \approx 0$).

One scenario of the form $\sigma_{\mathbf{Model}} < \sigma_{\mathbf{Image}}$

In our first scenario we increase the image noise by adding randomly sampled Gaussian

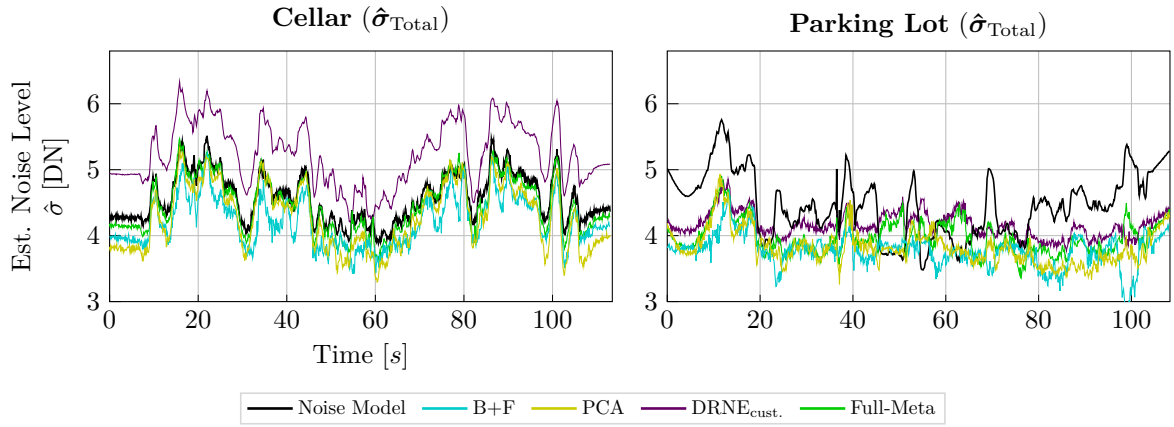


Figure 6.5: *Total noise estimation (datasets:* Cellar *and* Parking Lot*, camera:* ICX285*).* The left plot complements Fig. 6.4.

noise from $\mathcal{N}(\mu = 0, \sigma_{\mathcal{N}} = 5\,\text{DN})$ to the raw *Cellar* images. Note that this Gaussian noise is statistically independent from the other image noise sources and so its noise level adds in quadrature to the new total image noise level $\sigma_{\text{Image}+\mathcal{N}}$ (cf. (4.2)). We calculated the resulting ground truth $\xi_{\text{M/I}}$ as

$$\begin{aligned} \xi_{\text{M/I}} &\overset{(4.3)}{=} \sigma_{\text{Model}} - \sigma_{\text{Image}+\mathcal{N}} \\ &\overset{(4.2)}{=} \sigma_{\text{Model}} - \sqrt{\sigma^2_{\text{Image}} + \sigma^2_{\mathcal{N}}} \\ &\overset{(6.3)}{\approx} \sigma_{\text{Model}} - \sqrt{\sigma^2_{\text{Model}} + \sigma^2_{\text{Gauss}}}. \end{aligned} \tag{6.4}$$

The middle row of Fig. 6.4 illustrates the results. We expect only a reduction of $\xi_{\text{M/I}}$ and unchanged values otherwise, with respect to the first row. It can be seen that only *Full-Meta* captures the unexpected noise (note the initial error of $\approx 0.5\,\text{DN}$ is propagated), whereas *w/o-Meta* remains unchanged (cf. Sec. 6.3.1) 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* maintains its noise source estimated values.

Two scenarios of the form $\sigma_{\text{Model}} > \sigma_{\text{Image}}$

In this second test we increase the model noise by doubling the metadata *thermal white noise*. This parameter only affects *Full-Meta*. The new pseudo ground truth noise levels are calculated using the noise model. In a third test, we prepared an example with a doubled metadata *sensor temperature*, however, without new findings. Thus, it is treated in App. B.10.

The results are shown in the bottom row of Fig. 6.4. In this case, we expect an increasing $\hat{\sigma}_{\text{RN}}$ in accordance to the increased *thermal white noise*, an increasing $\hat{\xi}_{\text{M/I}}$ (which indicates the unexpected higher model noise) and unchanged values otherwise.

	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
	$\text{DRNE}_{\text{cust.}}$ + BM3D	43.01 / 0.9803	44.47 / 0.9853	45.53 / 0.9886	46.49 / 0.9913	47.59 / 0.9932
	w/o-Meta + BM3D	41.42 / 0.9671	43.83 / 0.9817	45.01 / 0.9861	46.26 / 0.9905	47.56 / 0.9930
	Min-Meta + BM3D	43.30 / 0.9818	44.72 / 0.9864	45.67 / 0.9899	46.49 / 0.9912	47.32 / 0.9922
	Full-Meta + BM3D	**43.74** / **0.9839**	**45.00** / **0.9875**	**45.99** / **0.9901**	**46.68** / **0.9916**	**47.63** / **0.9934**
	$\text{DRNE}_{\text{cust.}}$ + NLM	42.46 / 0.9793	43.91 / 0.9841	45.05 / 0.9874	46.09 / 0.9902	47.32 / 0.9925
	w/o-Meta + NLM	41.08 / 0.9701	43.33 / 0.9812	44.66 / 0.9322	45.94 / 0.9897	47.28 / 0.9924
	Min-Meta + NLM	42.77 / 0.9809	44.18 / 0.9852	45.18 / 0.9879	46.09 / 0.9902	46.92 / 0.9911
	Full-Meta + NLM	43.19 / 0.9828	44.47 / 0.9863	45.50 / 0.9889	46.26 / 0.9905	47.35 / 0.9927
	FBI-Denoiser	41.69 / 0.9830	42.07 / 0.9851	42.29 / 0.9865	42.41 / 0.9871	42.62 / 0.9880
	Blind2Unblind	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	35.24 / 0.9786
	$\text{DRNE}_{\text{cust.}}$ + BM3D	33.39 / 0.9546	33.78 / 0.9639	34.11 / 0.9713	34.51 / 0.9770	35.39 / 0.9820
	w/o-Meta + BM3D	33.04 / 0.9394	33.70 / 0.9612	34.07 / 0.9705	34.50 / 0.9770	35.37 / 0.9817
	Min-Meta + BM3D	33.47 / **0.9573**	33.83 / **0.9651**	34.11 / 0.9710	34.44 / 0.9749	35.20 / 0.9780
	Full-Meta + BM3D	33.40 / 0.9553	33.81 / 0.9648	**34.11** / **0.9715**	**34.51** / **0.9771**	**35.40** / **0.9823**
	$\text{DRNE}_{\text{cust.}}$ + NLM	33.14 / 0.9464	33.56 / 0.9567	33.92 / 0.9655	34.36 / 0.9731	35.29 / 0.9799
	w/o-Meta + NLM	32.93 / 0.7890	33.52 / 0.9559	33.91 / 0.9657	34.37 / 0.9738	35.26 / 0.9794
	Min-Meta + NLM	33.11 / 0.9447	33.52 / 0.9547	33.84 / 0.9622	34.17 / 0.9670	34.88 / 0.9709
	Full-Meta + NLM	33.14 / 0.9468	33.56 / 0.9566	33.92 / 0.9657	34.37 / 0.9734	35.31 / 0.9804
	FBI-Denoiser	32.52 / 0.9450	32.67 / 0.9522	32.76 / 0.9573	32.87 / 0.9604	33.05 / 0.9630
	Blind2Unblind	**33.61** / 0.9156	**33.86** / 0.9282	34.05 / 0.9379	34.27 / 0.9441	34.74 / 0.9488

Table 6.3: *Denoising performance for real-world images (camera: ICX 285).* Best PSNR (dB ↑) and SSIM (↑) 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).

We can see that *Full-Meta* meets these expectations (note the initial propagated error here as well).

Summary

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

6.4 Experiments on Real-World Image Denoising

The availability of camera metadata can improve the total image noise estimation (see Tabs. 6.1 and 6.2), which could also be beneficial for further downstream tasks besides camera readjustment, such as image denoising[2].

We investigate the effect of more accurate total noise level estimation on denoising on the example of two traditional denoisers that input expected noise levels (BM3D [Dab+07] and non-local means alias NLM [BCM05]) and compare results to two state-of-the-art learning-based denoisers (FBI-Denoiser [BCM21] and Blind2Unblind [Wan+22a]). BM3D and NLM both assume Gaussian noise, the FBI denoiser internally uses PGE-Net for Poisson-Gaussian noise estimation, and Blind2Unblind does not use explicit noise level

[2]Although we use the symptom-fighting application of denoising as a rationale for noise source estimation, denoising is the most studied application for noise level estimation in the literature and therefore best suited to assess the effects of estimating total noise more accurately.

representations, nor does it assume specific noise distributions. We apply all denoisers with default values and pre-trained weights provided by the respective authors (we select respective weights for real-noise images that lead 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 are applied to whole images. Denoising results are compared using peak signal-to-noise ratio (PSNR [dB]) and structural similarity index measure (SSIM) [Wan+04].

Table 6.3 presents quantitative results. For *Cellar*, *Full-Meta* + BM3D leads to the best results in all cases, followed by *Min-Meta* + BM3D for cases of higher noise, and $DRNE_{\text{cust.}}$ + BM3D for lower noise cases. We observe similar results in combination with the NLM denoiser. This is in accordance with the results from Tab. 6.1 that $DRNE_{\text{cust.}}$ and *Full-Meta* perform best and in accordance with Tab. 6.2 that *Min-Meta* is not far off, but it counteracts the non-intuitive results from Tab. 6.2 that *w/o-Meta* occasionally leads to more accurate total noise estimations. The denoising results rather underpin the intuition that the more metadata available for the noise source estimators, the better the total noise estimation (with the exception of the second best $DRNE_{\text{cust.}}$ in lower noise cases). The learning-based denoisers perform less accurate than BM3D, which differs 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 employ) nor fine-tuned to our datasets. The performance gap between the traditional and the learning-based denoisers increase with decreasing noise level (i.e., with increasing number of raw images used for averaging).

We note similar results for the *Parking Lot* dataset with the difference that Blind2Unblind and *Min-Meta* both perform best in the two highest noise cases (with respect to PSNR or SSIM, respectively). However, the better performance of *Min-Meta* compared to *Full-Meta* may be specific to BM3D, as *Full-Meta* is relatively more accurate when combined with NLM. Experiments with the *EV76C661* camera produce similar results and can be found in App. B.2.4.

Figure 6.6 illustrates qualitative results for both datasets using four averaged images as an example. It can be seen that the FullMeta+BM3D combination is the best at visually removing noise while preserving image detail, closely followed by DRNE+BM3D (e.g., FullMeta+BM3D restores the edges of the shadows less pixelated in the first row of Fig. 6.6). In contrast, FBI-Denoiser and Blind2Unblind visually remove noise best overall, but smooth the entire image (both, see especially bottom rows in Fig. 6.6) and introduce square artifacts (Blind2Unblind). NLM tends 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 Fig. 6.6).

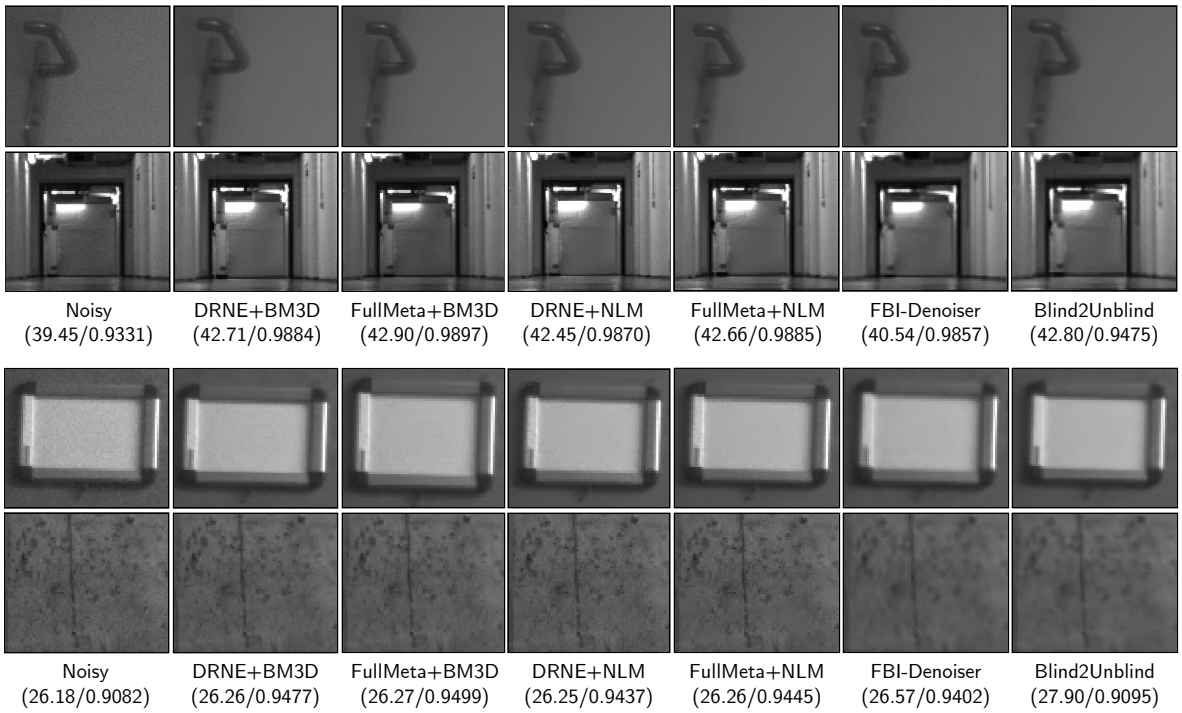


Figure 6.6: *Exemplary denoising results for real-world noised images (four averaged images, top rows:* Cellar*, bottom rows:* Parking Lot*).* Brightness and contrast are adapted for better visualization. FullMeta+BM3D best removes the noise while preserving image details. In contrast, FBI-Denoiser and Blind2Unblind remove noise best visually, but smooth the entire image (both) and introduce square artifacts (Blind2Unblind). NLM tends to retain noise at edges.

Summary

In summary, *Full-Meta* in combination with the traditional BM3D denoiser leads to the best denoising results in most cases. This supports previous findings that *Full-Meta* generally estimates the total noise level best and that noise estimation can benefit from camera metadata.

6.5 Camera Metadata Sensitivity Analysis

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

Let us first detail the experiment. We conduct a black box analysis by uniformly sampling different input values from their parameter spaces and observing respective outputs (i.e., the noise estimations). One input parameter is sampled at a time and other parameters are fixed to their respective maximum value (to aim for sufficiently high noise levels). Note that the only image feature the noise model depends on is the image intensity, while *Full-Meta* has learned to employ more image features (at least image noise). As we focus on the influence of camera metadata only, we only input uncorrupted homogeneous images with uniform intensities.

		Uniform samples of parameter value ranges									
		Min	1	2	3	4	5	6	7	8	Max
Noise Model	Mean Img 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
	Pixel Clock Rate*	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
	Thermal Wh. 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 Img 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
	Pixel Clock Rate*	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
	Thermal Wh. Noise**	9.46	9.54	9.71	9.87	10.0	10.1	10.1	10.1	10.1	10.1

Table 6.4: *Input-output sensitivity analysis of Full-Meta (bottom) compared to the noise model (top).* Input: One input parameter is sampled at a time while the rest are fixed to respective maximum values (worst case analysis) and the (uncorrupted) mean image intensity to 128 DN (to avoid saturation). Parameter value ranges are provided in Tab. 4.1 and concrete sampled values in Tab. B.3. Output: Estimated total noise level (table cells, in DN) per input parameter configuration. $\star$: 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. $\star\star$: Simulated CCD sensor (CMOS sensor otherwise).

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

We first examine the theoretical effect of each metadata on the estimated noise level according to the noise model. The top part of Tab. 6.4 depicts the results. 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, the thermal white noise, and the sense node reset factor, which all contribute to readout noise, have a negligible effect on the estimated noise level. Note that these three parameters are only insignificant for the considered set of fixed parameters in our experiments (cf. Tab. B.1). To illustrate, for instance, the influence of the pixel clock rate on the total noise level strongly depends on the correlated double sampling dominant time constant [KW14].

Comparing the top part with the bottom part in Tab. 6.4, the *Full-Meta* model has generally learned the relations between input parameters and the noise levels, but with some exceptions (colored values). Let us consider the two severe model deviations first (red values). The first differences are the estimated noise levels for minimum and maximum mean image intensities. This corresponds to the reduced noise estimation accuracy that we observed for under- and overexposed images (see App. B.2.3). The second deviation can be observed for full well capacities $\leq$ 24k electrons. These corresponding noise levels are most different from the other noise levels learned by *Full-Meta* (that range between about $[0, 13]$ DN). The farther the noise values depart from this range, the larger the observed model deviation. This indicates that these noise levels are underrepresented in the training data. Minor deviations from the noise model (orange values) are limited to small respective parameter values, with the exception of the sensor temperature. However, we do not see a specific pattern in these deviations; they are mostly slightly above 1 DN.

Summary

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 $\leq$ 24k electrons. The full well capacity and the camera gain could be identified as the most significant camera metadata, while pixel clock rate, sense nose reset factor, and thermal white noise could be neglected.

6.6 Discussion

In this section, we briefly address shortcomings and potential extensions of this chapter.

Camera Systems

We analyzed the proposed noise source estimators on two real-world camera systems, which show low noise statistics. However, we tried to investigate two more camera types (Realsense D435i RGB [Cor23] and Huawei P30 [Hua23] cameras), but we reached the point where camera manufacturers would only provide camera metadata for private usage behind a non-disclosure agreement.

Real-World Noise Extraction

The applied real-noise post-processing limits the real-world data quality because, although the real-world noise distribution is captured, the final noise is sampled from a rectified distribution in order to obtain unbiased noise with correct noise level labels. In addition, the generated temperature range and thus the induced DCSN is limited because we did not intend to damage the camera systems (we operated close to the suggested maximum working temperatures).

It is also worth noting that noise can be identified more precisely by measuring the currents at the respective places in the camera system processing chain (cf. for instance [Goi+10]). But this requires a laboratory environment with specialized equipment. Both are undesired for realistic experiments in the application domain.

Denoising Datasets

We provided datasets with 50 frames per setting, which might limit the reliability of the results, but similar to the blur estimation datasets, we kept the size small to limit the computation time required for traditional denoising. Future work could consider large-scale experiments.

Abdelhamed et al. [ALB18] further addressed two common shortcomings when creating real-world denoising datasets with ground truth data obtained by image averaging:

(*i*) GT images may be blurry if there is misalignment between images in the averaged image sequence (e.g., in non-static scenes), and

(*ii*) GT images may be biased if the averaged noise has non-zero-mean (e.g., for clipped pixel intensities in under- and overexposed areas).

Scene misalignment can be tackled by image registration [Zha+19]. In contrast to static scenes, registration increases the dataset variability, but at the cost of registration inaccuracies and thus GT data quality. Regarding the second shortcoming, we observe overexposure for the *ICX285 Parking Lot* dataset (near light tubes) and for the *EV76C661 Parking Lot* dataset (large image areas overexposed from illumination outside the parking garage). Both limit the quality of the datasets, but are unavoidable for the application of traditional cameras to real scenes with high dynamic range.

Noise Source Estimator Architecture

Runtime tests indicate that the PGE-Net's U-Net architecture performs significantly faster compared to a standard CNN architecture, which could be investigated to improve the noise source estimators. Further choices for combining information from image and metadata are detailed in Sec. 2.3.2 (e.g., encode metadata in additional image channels).

Camera Metadata Sensitivity Analysis

We tailored our analysis to uncorrupted, homogeneous images, but the noise source estimators could have learned useful relations between metadata and image quality attributes. Future work could examine more attributes, such as texture or contrast (more details on image attributes in Sec. 3.2).

According to the sensitivity analysis results in Tab. 6.4, we also propose a retraining of the noise source estimators, when working with cameras whose full well capacities $\leq$ 24k electrons. Note this also holds for the camera systems we employed (cf. Tab. A.1). However, we would like to point out that we inspected *maximum* noise levels in the sensitivity analysis and that the effect of the full well capacity parameters on the noise estimation can be expected to be smaller in most cases.

Another more common approach to analyze the sensitivity of DNN's input data is "SHapley Additive exPlanations" (SHAP) [LL17], but this procedure is not yet applicable to multiple input branches with heterogeneous shapes [Joh23].

6.7 Summary

In this chapter, we evaluated the idea of noise source estimation from camera metadata and a captured image to determine the noise contribution of different noise sources from within a camera system. Noise source estimation is considered the second part of our proposed condition monitoring module.

We first proposed (*i*) four datasets with ground truth labels and (*ii*) two datasets with pseudo ground truth labels (Sec. 6.1). The former datasets (*i*) were augmented with both synthetic noise generated by the theoretical noise model (using randomly sampled camera metadata) and real-world noise extracted from two camera systems. The two datasets with pseudo ground truth (*ii*) were acquired in real-world field campaigns and include unprocessed real-world noise from the same two cameras. Multiple static image sequences from (*ii*) are further selected and processed for denoising experiments.

From the quantitative results in Sec. 6.2, we found that only the model with access to the full set of camera metadata (*Full-Meta*) could accurately and robustly quantify the contribution of each noise source. Although all variants of the proposed method can estimate total noise well, the lack of camera metadata for *w/o-Meta* and *Min-Meta* makes it difficult to disambiguate the different noise sources. We further demonstrated that the incorporation of metadata increases the inference time marginally.

Additional experiments on the real-world platforms (Sec. 6.3) supported the quantitative results and further underpinned the superiority of the *Full-Meta* noise source estimator.

In Sec. 6.4, we investigated a potential positive impact from the improved total noise estimation on the downstream task of image denoising by comparing two traditional denoisers (coupled with our noise source estimators) with two modern learning-based denoisers on quantitative and qualitative denoising results. Combining *Full-Meta* with the traditional BM3D denoiser yielded the best results and provided additional evidence that *Full-Meta* estimates the noise level best among the tested noise estimators.

In a final experiment (Sec. 6.5), we analyzed the effect of each camera metadata on noise level estimation by comparing *Full-Meta* with the theoretical noise model. We found that *Full-Meta* learned to capture the theoretical camera metadata relations with notable exceptions for potentially under- and overexposed images, and large noise levels resulting from camera full well capacities $\leq$ 24k electrons. The full well capacity and the camera gain were identified as the most significant camera metadata, while pixel clock rate, sense nose reset factor, and thermal white noise were identified as the least significant, with a negligible effect.

Lastly, we addressed shortcomings and potential extensions in Sec. 6.6. Among others, we discussed means to improve data quality of extracted real-world noise (offset control, noise measurement on a current level), potential improvements for noise source estimation (U-Net architecture, other combinations of image and metadata), and further experiments on camera metadata analysis (more image quality attributes). Moreover, we referred to difficulties on testing additional camera systems (undisclosed metadata) and addressed common pitfalls for creating denoising datasets from which ours are also partially affected (under- and overexposed images lead to biased ground truth).

CHAPTER 7

Evaluation: Integrated Self-Health-Maintenance Framework

In this chapter we evaluate the combination of online blur/ noise estimators and offline empirical input-output performance curves (IOPCs) for the practical application to control image quality and hence optimize the system's performance. In Sec. 7.1, we first determine exemplary IOPCs that relate object detection performances to blur and noise effects of different severities. We then propose a simulated and a real-world scenario (Sec. 7.2) on which we demonstrate the application of the IOPCs (Sec. 7.3). Subsequently, the framework's required computational cost on stationary and mobile hardware is investigated in Sec. 7.4. Finally, this chapter is concluded with a discussion (Sec. 7.5) and a summary (Sec. 7.6). This chapter is partially published in [Wis+23b].

7.1 Object Detection Sensitivity Analysis

We choose object detection as our exemplary target application and specifically examine YOLOv4 and Faster R-CNN (cf. Sec. 4.4.1) for the object classes "car" and "pedestrian" on Udacity data. Our focus lies on actions tackling linear motion blur (LinMB) here because object detectors are substantially more sensitive to LinMB than to noise (Figs. 7.1 and 7.2), and there is abundant motion blur in standard datasets like Udacity (Fig. 5.6). We also neglect photon shot noise and demonstrate the procedure with sensor noise only (dark current shot noise plus readout noise), so that filtered photon noise does not lower the noise level estimation of simultaneously occurring blur and noise[1]. Unless otherwise stated, we apply the settings from Sec. 5.1.1 for blur and noise generation. Furthermore, the improved blur estimation presented in Sec. 5.5 is used to ensure better estimations in presence of high noise.

Figure 7.1 shows exemplary IOPCs for isolated and Fig. 7.2 for combined blur and noise occurrences. It can be seen that both object detectors and object classes are each affected differently by isolated and combined blur and noise.

[1]Note that photon shot noise in images can be mitigated in reality, for instance, by using a camera with a large full well capacity and a well-illuminated scene (i.e., a strong signal in the image).

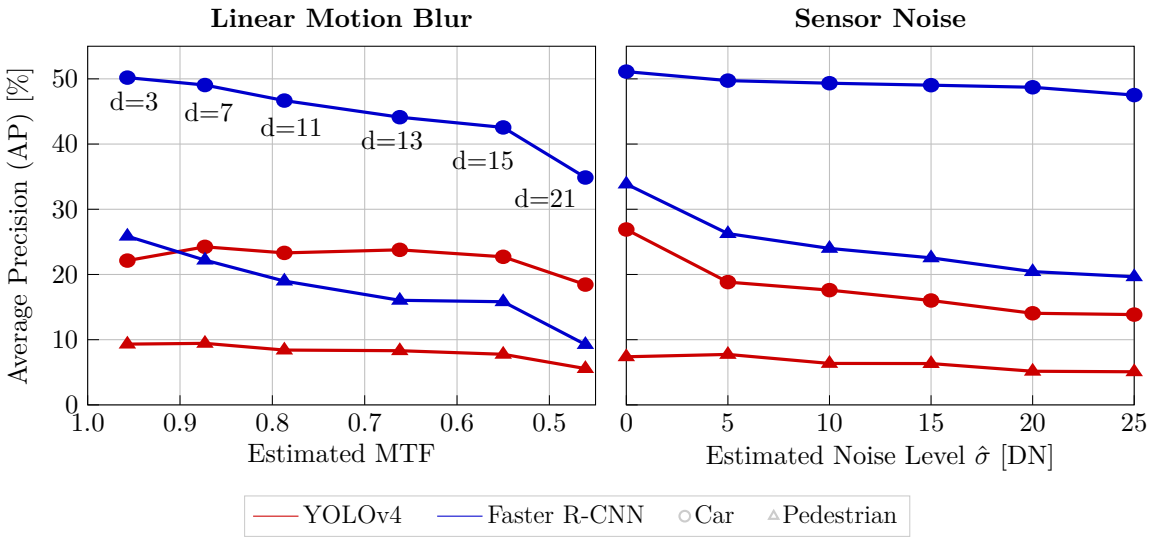


Figure 7.1: *Influence of isolated blur and noise on object detection performance.* Both input-output profiles depend on the actual estimated corruption levels. Noise levels and MTFs (blur sizes d [px] in black) are estimated by the respective CNN methods and the MTFs depict means for horizontal and vertical measurements at frequency $f = 0.1$ (averaged for four motion blur directions). Object detection performances are measured in terms of average precision (AP).

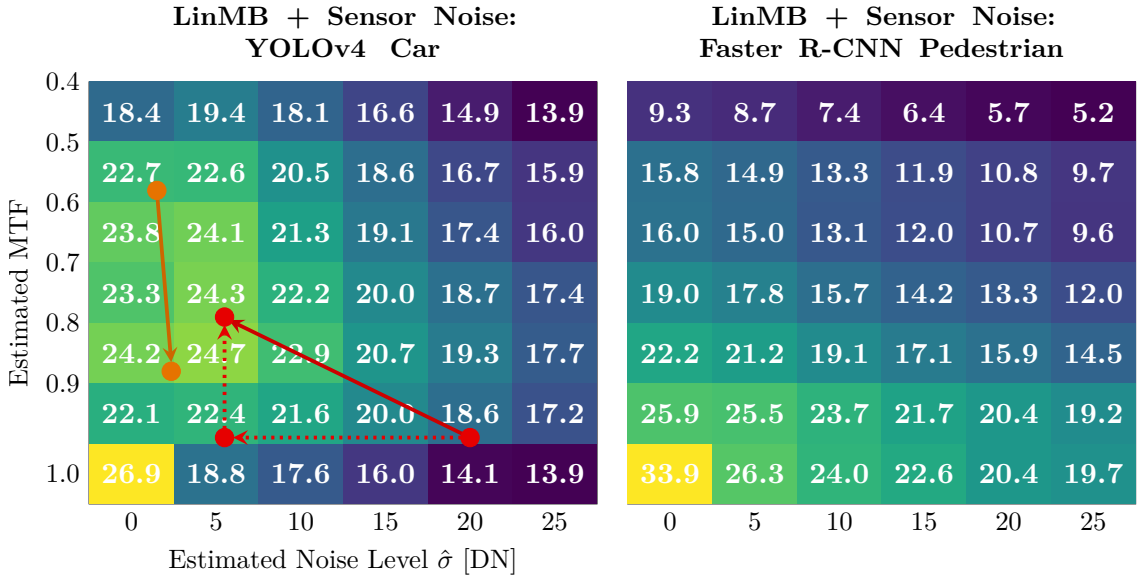


Figure 7.2: *Influence of combined blur and noise on object detection performance.* Compare to their one-dimensional counterparts in Fig. 7.1. The red and orange arrows demonstrate two examples of exposure time t_{exp} / ISO-gain trade-off paths (see text in Sec. 7.3).

Moreover, the relation from the blur or noise corruptions to the detection performances might be non-trivial and non-linear (e.g., YOLOv4 car detection in presence of LinMB), since we do not really know what machine learning methods learn.

7.2 Datasets

We demonstrate the framework on the example of YOLOv4 car-detection in a modified *Sim* scene and in a real-world *Parking Lot* scene.

Sim

The following augmentation steps are applied to the uncorrupted *Sim* scene (cf. Sec. 5.1.1):

(*i*) *Linear Motion Blur and Sensor Noise Corruptions.*
To generate blur, we include camera movement of a real-world trajectory (extracted from a navigation module of a real *ICX285* camera [Bör+17]) with a speed of $v \approx 760\,\mathrm{px/s}$ at the distance of the car objects and set the simulated exposure time to $t_{\mathrm{exp}} = 4\,\mathrm{ms}$. This causes linear motion blur of size $d = v \cdot t_{\mathrm{exp}} = 0.76\,\mathrm{px/ms} \cdot 4\,\mathrm{ms} \approx 3\,\mathrm{px}$. In the simulation, the blurred image is realized by sampling and averaging 100 frames per exposure time span (similar to [CFZ19]).

To generate noise, we apply our noise model (Sec. 3.2.2) with parameters from the *ICX285* camera (Tab. A.1), a camera temperature of $T = 330\,\mathrm{K}$, and raw noise amplification to reach $\sigma = 20\,\mathrm{DN}$. As for the IOPCs, photon shot noise is omitted (cf. Sec. 7.1).

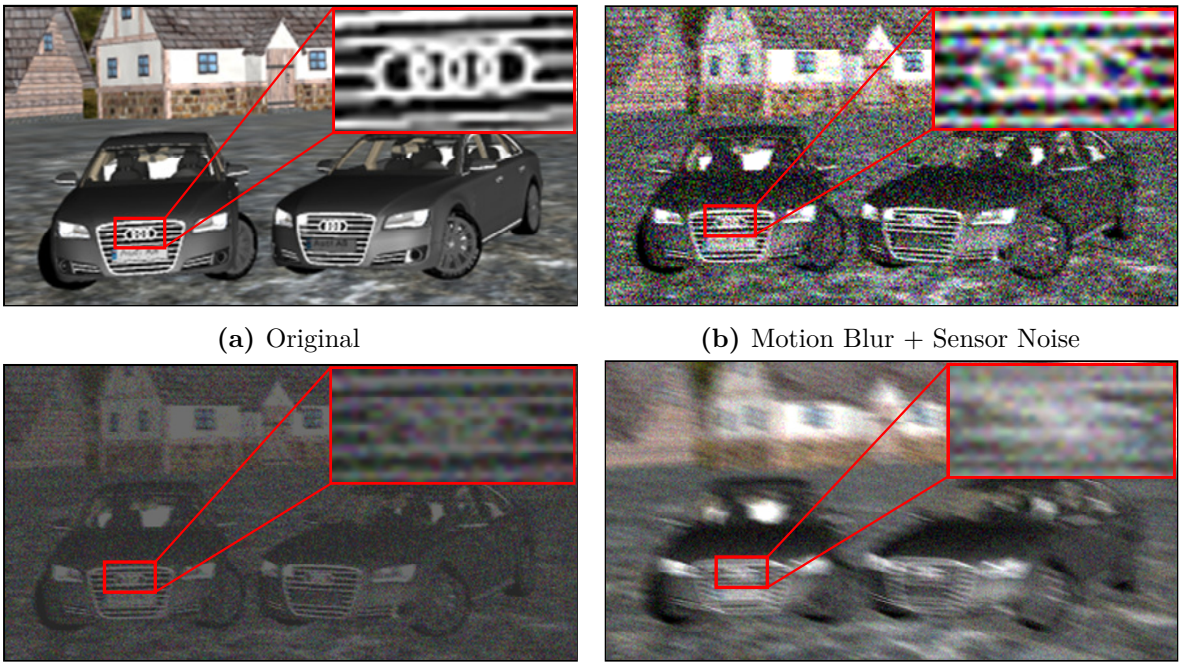
(*ii*) *ISO Gain Alteration.*
We first estimate noise and blur statistics from the images simulated in (*i*) and apply these to our YOLOv4 car-detection IOPC (left plot of Fig. 7.2). The result is a factor $\alpha^\star$ (4.7) by which we alter the simulated camera ISO gain, which is implemented as a noise and image intensity amplification factor.

(*iii*) *Exposure Time Compensation.*
As we did not investigate the influence of image intensity on object detection performance, we subsequently alter t_{exp} (and the simulated intensity amplification factor) by the factor $\alpha^\star$ to restore the original intensity level.

Finally, together with the uncorrupted *Sim* images, we obtain four *Sim* sub-datasets (Fig. 7.3).

Parking Lot

This dataset depicts the same scene as the eponymous dataset from Sec. 6.1.2. The images are recorded by navigating our camera system on the robotic platform (Fig. 5.4a) through the low-illuminated parking garage with fixed initial $t_{\mathrm{exp}} = 8\,\mathrm{ms}$, ISO = 100, and default camera parameter values for the rest (Fig. 7.4a). To make the car-detection performance curve applicable, we target (*i*) a constant linear motion blur induced by a constant speed of the platform and (*ii*) a low noise level with a low initial ISO gain



(a) Original **(b)** Motion Blur + Sensor Noise

(c) ISO Gain Alteration **(d)** Exposure Time Compensation

Figure 7.3: *Self-health-maintenance framework evaluation on synthetic data (*Sim*).* (a): Uncorrupted *Sim* scene. (b): Linear motion blur and sensor noise are added to the scene ($t_{\text{exp}} = 4\,\text{ms}$, ISO $= 400$). (c) and (d): Exemplary camera ISO gain reduction and exposure time increase that resulted from applying blur and noise estimations to an object detection performance profile in order to maximize car detection performance ($t_{\text{exp}} = 16\,\text{ms}$, ISO $= 100$).

(a) Fixed Cam. Parameters **(b)** Default Cam. Control **(c)** Proposed Cam. Control

Figure 7.4: *Self-health-maintenance framework evaluation on real-world data (*Parking Lot*).* Brightness and contrast are adapted for better visualization. (a): Fixed default exposure time (t_{exp}) and ISO gain camera parameters ($t_{\text{exp}} = 8\,\text{ms}$, ISO $= 100$). (b): Parameters adjusted by default camera control ($t_{\text{exp}} = 17\,\text{ms}$, ISO $= 100$). (c): Parameters adjusted by proposed camera control ($t_{\text{exp}} = 3.7\,\text{ms}$, ISO $= 224$).

to mitigate the undesired impact of photon shot noise. Analogously, the experiment was repeated with the camera's built-in t_{exp} / ISO gain controller [Gmb21] enabled (Fig. 7.4b) and with our proposed parameter controller (Fig. 7.4c). The camera parameter controllers were automatically triggered on the first image frame at $t = 0\,\text{ms}$. Finally, we sampled the video sequences for each configuration such that each one contains 200 ± 5 cars, which we manually annotated for YOLOv4 object detection.

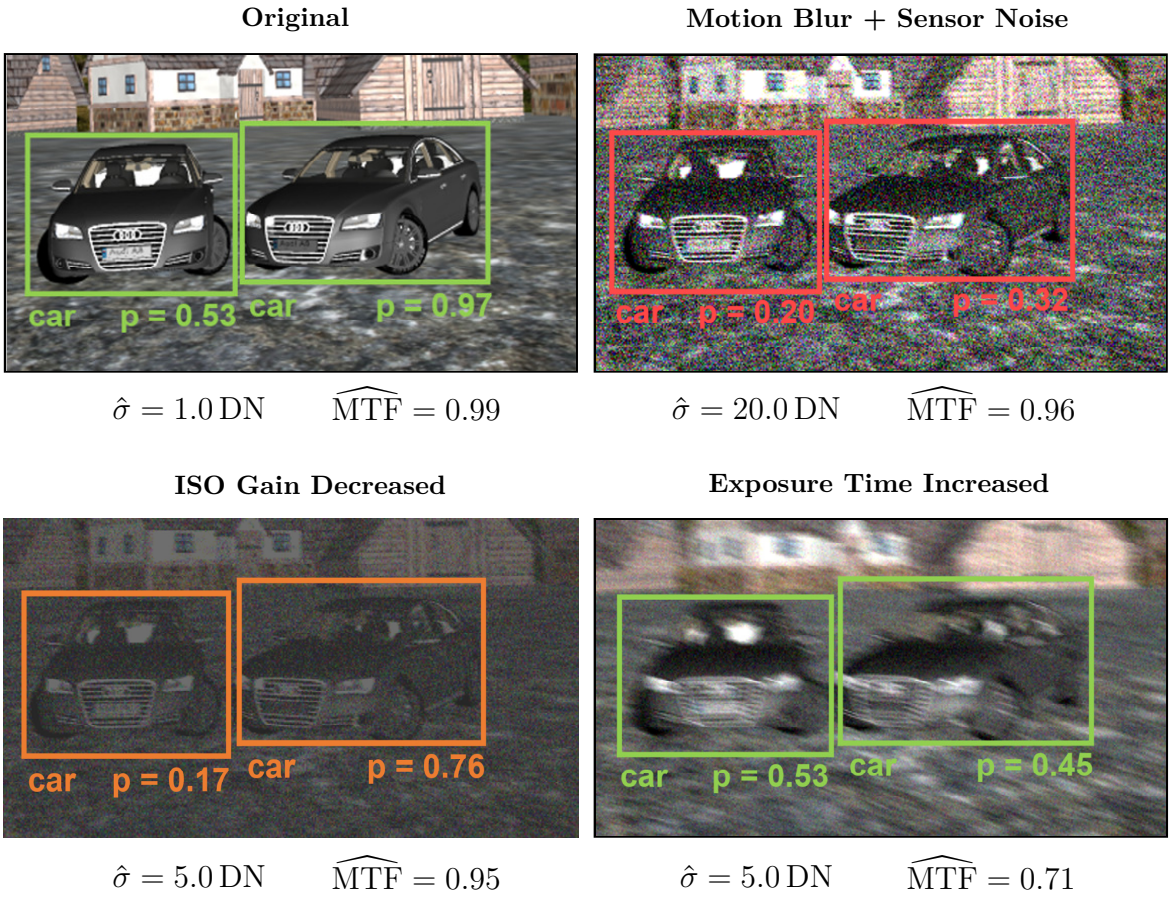


Figure 7.5: *Maximizing object detection by trading off blur and noise (*Sim*).* Application of the proposed framework to detect cars using YOLOv4 on *Sim* data suffering from linear motion blur and sensor noise. The original scene is first imaged with an ISO gain of 400 (leading to sensor noise of $\sigma \approx 20\,\mathrm{DN}$) and an exposure time of 4 ms. As a result, the car recognition performance of YOLOv4 decreases drastically (top-right image). Applying the optimal $\alpha^\star \approx 0.25$ (according to the performance profile from Fig. 7.2) improves car detection (bottom-left image). Finally, we divide the exposure time by $\alpha^\star$ to compensate for the missing light, which improves overall detection slightly (bottom-right). Hence, noise is reduced from $\sigma \approx 20\,\mathrm{DN}$ to 5 DN and blur is increased from $d \approx 3\,\mathrm{px}$ to 12 px while the average detection rate increases from $\bar{p} \approx 0.26$ to $\bar{p} \approx 0.49$.

7.3 Optimizing Object Detection by Trading off Blur and Noise

We first demonstrate the functionality of our framework on the *Sim* dataset and thereafter evaluate the framework integrated in the *ICX285* camera system on the real-world *Parking Lot* dataset.

Sim

We apply this framework in the *Sim* environment on a concrete example of YOLOv4 car detection with data corrupted by linear motion blur and sensor noise (Fig. 7.5). The first image in Fig. 7.5 depicts the scene in uncorrupted conditions (without noise or blur), for reference.

Here the first car is detected fairly ($p = 0.53$) and the second one much better ($p = 0.97$). While the CNN noise estimator detects a small noise level of $\hat{\sigma} = 1\,\mathrm{DN}$ by mistake, the MTF estimation is nearly ideal ($\widehat{\mathrm{MTF}} = 0.99$). Subsequently, we applied motion blur and sensor noise ($d = 3\,\mathrm{px}$ and $\sigma = 20\,\mathrm{DN}$). In this situation (top-right image in Fig. 7.5), blur and noise are estimated within the expected error ranges ($\hat{d}_{\mathrm{old}} \approx 3\,\mathrm{px}$ and $\hat{\sigma} = 20\,\mathrm{DN}$), but the cars are detected worse on average ($\bar{p} \approx 0.26$).

In the next step, we determine $\alpha^\star$: knowing the relation between motion blur sizes and estimated MTFs (first plot in Fig. 7.1) and the estimated noise level, we target an $\widehat{\mathrm{MTF}} \in [0.7, 0.8]$, which corresponds to approximately $\hat{d} \in [11, 12]\,\mathrm{px}$ (cf. first plot in Fig. 7.2). We choose $d_{\mathrm{target}} \approx 12\,\mathrm{px}$, hence, $\alpha^\star = \hat{d}_{\mathrm{old}}/d_{\mathrm{target}} \approx 3\,\mathrm{px}/12\,\mathrm{px} = 0.25$. We then reduce the ISO gain by the factor $\alpha^\star$ and show an intermediate image without increasing t_{exp} (bottom-left image of Fig. 7.5). One car is now detected more confidently while blur and noise are still estimated within the expected error ranges. Following, we increase t_{exp} by the factor $\alpha^\star$ to restore the original intensity level, producing the bottom-right image of Fig. 7.5. In this last step, the total detection score slightly increases ($\bar{p} \approx 0.49$) despite the likewise motion blur amplification ($d \approx \hat{d} = 12\,\mathrm{px}$). The steps taken are marked with red arrows on the heat plot in Fig. 7.2.

Parking Lot

For this real-world example, we employed a mobile computer doing the real-time calculations (CPU: Intel i7-9850H, GPU: NVIDIA MX150). With this setup, we demonstrate another example of the non-monotonic YOLOv4–car-detection heat map of Fig. 7.2, marked with an orange arrow. The results are shown in Fig. 7.6.

The built-in camera controller tracks a mean image intensity level of 50% and prioritizes changing t_{exp} over ISO gain as long as $t_{\mathrm{exp}} \leq 500\,\mathrm{ms}$. Hence, the built-in controller constantly changed t_{exp} only, did not account for the motion blur, and resulted in an AP car detection score of 26.08%.

Our proposed framework took about 3 s to estimate $(\hat{\sigma}, \widehat{\mathrm{MTF}}) = (0.1, 0.57)$ (longer than in Ch. 5 due to the weaker mobile hardware, but still interactive / real-time). With initially fixed camera parameters (i.e., while $t < 3\,\mathrm{s}$), YOLOv4 reached an AP score of 47.54%. The system then decided to decrease the motion blur at the expense of slightly increasing the noise to move to higher AP detection values (brighter part of the heat map). Inspecting the AP curves (first plot in Fig. 7.1), $\widehat{\mathrm{MTF}} = 0.57$ corresponds to $\hat{d} = 15\,\mathrm{px}$, and the system targeted $\widehat{\mathrm{MTF}} \in [0.8, 0.9]$ (high values of the heat map), which corresponds to a smaller motion blur of $\hat{d} \approx 7\,\mathrm{px}$. Two steps were taken: first, the system decreased the exposure time by a factor $\alpha \approx 15\,\mathrm{px}/7\,\mathrm{px} \approx 2.14$ to achieve the desired MTF improvement. Then, it increased the ISO (and increased noise) by the same factor $\alpha \approx 2.14$ to restore the intensity level for the detector. The final operating point was $(\hat{\sigma}, \widehat{\mathrm{MTF}}) \approx (0.4, 0.9)$, which has a higher AP value (60.56%) than the initial point.

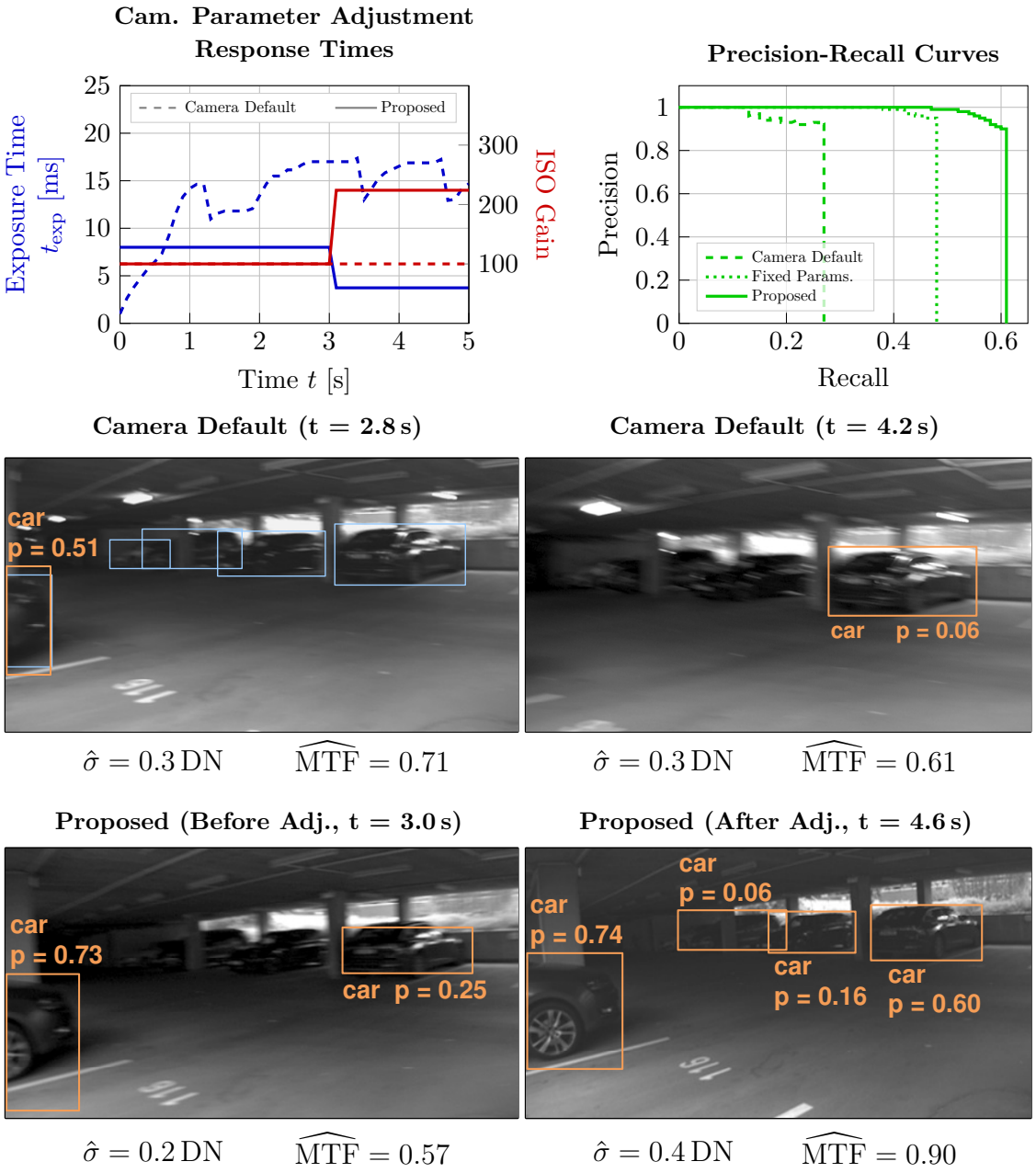


Figure 7.6: *Comparison of built-in camera t_{exp} / gain control vs. our framework (*Parking Lot*).* Left plot: The built-in controller constantly optimized image intensity by adjusting t_{exp} only. In contrast, our framework targets optimal object detection performance and adjusted t_{exp} / gain once. Right plot: Precision-Recall curve that details object detection performances for the three tested camera parameter configurations. Images: Examples from the experiments (adapted brightness and contrast for better visualization). Blue boxes indicate ground truth objects, and orange boxes actual detections. The overall AP scores are: 26.08% for the built-in camera control, 47.54% for the manually chosen fixed parameters (before automatic adjustment), and 60.56% for our framework (after automatic adjustment).

The precision-recall curves corresponding to the measured AP scores are depicted in Fig. 7.6, for reference – the larger the area under a curve, the better the detection performance (cf. AP definition in Sec. A.1).

Summary
We were able to demonstrate that an offline-learned synthetic input-output performance profile could be applied to a simulated and a real-world scenario of the same domain in real-time to improve car-detection accuracies on the basis of online image blur and noise estimation.

7.4 Computational Cost

In this section, we analyze the computation time and working memory that the proposed framework requires on stationary and mobile hardware at inference time. As stationary system, we use an Intel Xeon W-2145 CPU with 64 GB working memory and an Nvidia RTX Quadro 6000 GPU with 24 GB dedicated working memory, and perform actual performance measurements. As mobile hardware, we consider an "NVidia Jetson AGX Orin" AI-module that is equipped with an Arm Cortex-A78AE v8.2 CPU and 64 GB shared working memory [Kar22], on which we calculate expected performances.

Pre-Considerations
The framework's total computational cost is composed of the individual costs from its (*i*) *target application*, (*ii*) *image attribute (source) estimators*, and (*iii*) *decision & control unit*. We make five considerations for the following evaluation:

1. We examine the target application as part of the framework, since resources are shared at inference time.
2. The faster and more lightweight YOLOv4 is considered as (*i*) target application, and only the ML-based methods for (*ii*) image attribute estimation.
3. We follow the literature and express computational cost of ML-based methods in terms of required (single-precision) floating point operations per second (FLOP/s) [Ma+18].
4. ML-based models are executed on GPU and data pre-/ post-processing on CPU. Computation time is determined with built-in Python functions, occupied working memory using a memory profiler [Ped+22], and required GPU FLOP/s with the tool of [tok20]. We provide computation times for a whole image (1280×1024 px) in the case of YOLOv4 and otherwise per image patch (patch size determined by the respective method). All tests are executed three times and the results are averaged to counteract the influence of background processes.
5. We neglect the computational cost of the (*iii*) decision & control policy, as it uses an offline-calculated look-up-table at inference time that requires only memory in the magnitude of kilobytes and a constant small access time per lookup operation.

Stationary Hardware

Table 7.1 summarizes the results measured with the stationary system. Let us first consider each processing stage separately. In the pre-processing, only YOLOv4 and the blur estimator perform time-consuming image operations (e.g., image resizing and sobel filtering, respectively). The loaded model sizes do not stand out, considering working memory capacities in the order of GB nowadays. In the inference phase, all methods need comparably low computation time, with YOLOv4 and the noise estimator taking the longest. YOLOv4 requires comparably less GPU memory by having the largest number of FLOP/s because the YOLO approach has matured over eight years since its first version. Lastly, all methods post-process their data resource-efficiently.

When it comes to the overall computational cost, we distinguish between sequential and concurrent execution of all components to assess overall individual performances as well. In the sequential part, the blur estimator and YOLOv4 require the most computation time. On the other hand, the noise and noise source estimators require around the double amount of GPU working memory during inference. However, all methods are, with a total runtime of 89.4 ms or 11.2 frames per second, still executable in real-time and within the available memory limit (1.8 GB CPU and 15.8 GB GPU working memory allocation). In the concurrent part, we observe that our proposed framework (FW) benefits from parallel execution, which almost halves its execution time to $t_{\mathrm{FW;conc}} = 50.4\,\mathrm{ms}$ (19.8 frames per second) at the expense of more GPU memory overhead (19.8 GB GPU working memory allocated).

Figure 7.7 illustrates the CPU and GPU loads during the concurrent framework execution. It shows that the CUDA processing capacities could be a potential bottleneck for future framework extensions (GPU CUDA cores load constantly at about 86%) and further underpins that CPU data pre- and post-processings benefit from parallel execution (all CPU cores utilized). However, the cores are not used to capacity, although pre- and post-processing on the CPU dominate the total required runtime. This indicates that there is still capacity for improvement in terms of parallel calculations.

Mobile Hardware

Let us now consider the mobile Orin platform. We distinguish between (*i*) *working memory*, (*ii*) *GPU* and (*iii*) *CPU computation capabilities*:

(*i*) *Working Memory:* The Orin module provides up to 64 GB of shared random access memory [Kar22]. We determined a total working memory requirement of $19.8 + 1.8 = 21.6 < 64\,\mathrm{GB}$ (Tab. 7.1).

(*ii*) *GPU:* The Orin module further theoretically enables up to 5.3T FLOP/s on CUDA cores [Kar22]. We measured a requirement of $91.3\,\mathrm{B} < 5.3\,\mathrm{T}$ GPU FLOP/s in

Process	Time [ms]	CPU Mem. [MB]	GPU Mem. [GB]	GPU FLOP/s [B]
Pre-Processing				
Blur Estimator	36.1	538.4	-	-
Noise Estimator	0.2	476.7	-	-
Noise Source Estimator	0.2	414.8	-	-
YOLOv4	15.6	375.3	-	-
Inference				
Blur Estimator	1.4	-	2.6	3.4
Noise Estimator	15.6	-	5.4	17.0
Noise Source Estimator	1.4	-	5.2	10.9
YOLOv4	15.7	-	2.6	60.0 [PA21]
Post-Processing				
Blur Estimator	-	-	-	-
Noise Estimator	0.1	0.1	-	-
Noise Source Estimator	1.1	0.6	-	-
YOLOv4	10.0	3.4	-	-
Overall (Sequential)				
Blur Estimator	37.4	538.4	2.6	3.4
Noise Estimator	10.7	476.7	5.4	17.0
Noise Source Estimator	2.6	414.8	5.2	10.9
YOLOv4	38.7	375.3	2.6	60.0
Overall (Concurrent)	50.4	1805.2	19.8	91.3

Table 7.1: *Computational cost measurements for the proposed self-health-maintenance framework running on the stationary system.* Computation times are calculated per image (YOLOv4) or image patch (estimators). CPU and GPU memory values denote peak allocated memory.

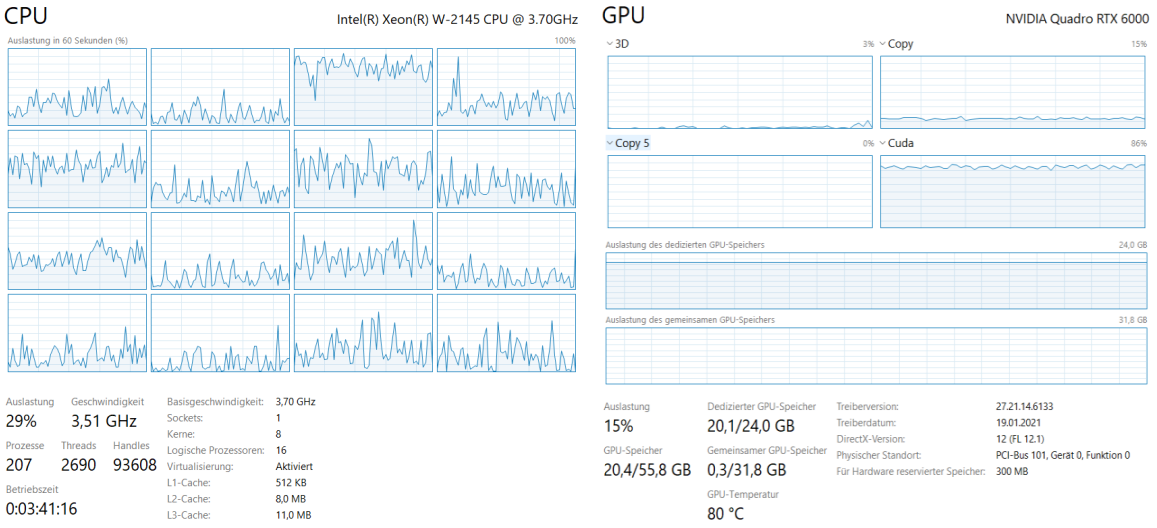


Figure 7.7: *Total GPU and CPU loads during concurrent framework execution* visualized by the task manager of the Windows 10 operating system.

concurrent operation (Tab. 7.1), which yields up to 58 frames per second (17.23 ms per frame).

(*iii*) *CPU:* The Orin module is also equipped with an Arm Cortex-A78AE v8.2 CPU with 12 cores and up to a 2.2 GHz clock rate [Kar22]. To compare this CPU to

the Intel Xeon W-2145, we perform the following steps: (a) measurement of peak FLOP/s that the Intel CPU can execute, (b) estimation of FLOP/s needed for the framework execution using (a), and (c) estimation of peak FLOP/s that the Arm CPU can execute as well as its execution time for the framework using (b).

(a) From the single-precision general matrix multiply (SGEMM) benchmark [Yin17] (with matrix size 192×192 that matches our blur estimation patch size), we obtain a peak FLOP/s performance of $\text{FLOP/s}_{\text{Peak;Intel}} = 158.82\text{G}$ FLOP/s for the Intel CPU.

(b) On the basis of the measured framework execution time $t_{\text{FW;conc}} = 50.4\,\text{ms}$ in concurrent operation (Tab. 7.1), we estimate a computation time requirement to execute the framework of $\text{FLOP}_{\text{FW;Intel}} = \frac{\text{FLOP/s}_{\text{Peak;Intel}} \cdot t_{\text{FW;conc}}}{1000} = \frac{158.82 \cdot 50.4}{1000} = 8\text{G}$ FLOP.

(c) The specification of the Arm CPU lists an execution throughput of 2 instructions per cycle for "FP arithmetic" for "AArch64 FP" and "AArch32 FP" instructions [Lim21, p. 14, pp. 27–28]. On this basis, we can now estimate the Arm's theoretical peak FLOP/s following [Dol15][2]:

$$\begin{aligned} \text{FLOP/s}_{\text{Peak;Arm}} &= \text{Max. Clock} \times \#\text{Cores} \times \text{FLOP per Cycle} \\ &= 2.2\,\text{GHz} \times 12\ \text{Cores} \times 2\ \text{FLOP per Cycle} \\ &\approx 52.8\,\text{G FLOP/s}. \end{aligned} \tag{7.1}$$

Putting all together, the Arm CPU can theoretically execute the CPU instructions of the framework with up to $t_{\text{FW;Arm}} = \text{FLOP/s}_{\text{Peak;Arm}} / \text{FLOP}_{\text{FW;Intel}} = 52.8/8 = 6.6$ frames per second (i.e., it requires a computation time of approximately 151.52 ms).

All in all, the framework could be executed with a maximum of 6.6 frames per second on the mobile Orin module. Comparing the performance statistics from (i) – (iii), the CPU could be considered as the bottleneck, while working memory and GPU performance are both sufficiently available.

Performance Improvement Considerations

The computational cost of the framework can still be significantly reduced:

General: The total computational cost scales with the number of used image quality attribute (source) estimators. The more image attributes are estimated at inference time, the larger the total resource requirements.

[2] We conservatively assume the values for $\frac{\text{FLOP/s}}{\text{operation}}$ and $\frac{\text{operations}}{\text{instructions}}$ as equal to one as "Each issue pipeline can accept one micro-operation per cycle." [Lim21, p. 11], which is plausible compared to values from [Dol15]. For the sake of simplicity, these terms are therefore omitted in the equation.

CPU time: The main bottleneck of the Orin module results from CPU computations, i.e., blur estimator and YOLOv4 pre- and post-processings, which could be vectorized or learned as part of the respective CNNs to run on the GPU.

GPU time and memory: Latest object detectors such as YOLOv8s [JCQ23] could be employed, which the authors claim to perform similarly accurate and to require half the FLOP/s of YOLOv4. There are also dedicated mobile models such as YOLOv8n [JCQ23] that further reduce the required computational resources at the expense of detection accuracy.

GPU time and memory: The required working memory can be reduced if network parameters are expressed in terms of smaller data types. Currently, the estimators use 32-bit floating point numbers that can be reduced to 16-bit ones or 8-bit integers. Depending on the used processing unit and the data type, the computation time capability might increase (e.g., for the NVidia Jetson AGX Orin from 5.3T FLOP/s to up to 275T[3] OPS using 8-bit integers [Kar22]). However, the respective data type must be supported by the processing unit.

Summary

In summary, the real-time capability of the framework could be demonstrated on a stationary system (19.8 frames per second) and calculated for a mobile AI platform (6.6 frames per second). These numbers could be improved by using one of the latest YOLOv8 models, 8-bit integer precision for CNN parameters, or pre- and post-processing as part of the CNNs.

7.5 Discussion

Here we briefly discuss further details and shortcomings of the object detection sensitivity analysis results, the practical application of the performance profile, and the computational cost analysis of the integrated framework.

Object Detection Sensitivity Analysis

The resolution of a performance profile (i.e., its bin size) is limited by the accuracies of the used image quality attribute estimators. Moreover, changing a bin size is a trade-off with the amount of data to be analyzed in the sensitivity analysis. Since the sensitivity is analyzed only once offline (and can be well parallelized) and the resolution does not affect the profile's inference time, maximum resolution of the performance profile should be preferred.

[3]Use this number with caution. It is neither specified how this value is obtained, nor what the term "AI performance" they use refers to.

In the specific example of the linear motion blur and sensor noise profiles (Figs. 7.1 and 7.2), the bin resolution of the sensor noise axis could be increased to around 3 DN (cf. results from Sec. 5.4), whereas the motion blur bin resolution is with 0.1 approximately at its measured maximum (cf. results from Sec. 5.5).

Note that we used averaged horizontal and vertical MTF measurements at frequency $f = 0.1$ lines/px as blur indicator. The use of higher frequencies (i.e., focusing on higher image details) could be of greater importance when dealing with lower blur levels, as it enables better separation of the blur kernels and thus increase the robustness of the detection. In addition, the performance profile could be split for each MTF direction to allow more specific countermeasures.

Optimizing Object Detection by Trading off Blur and Noise

We noticed a significant effect of image intensity on object detection performance (cf. Fig. 7.5), which complements our observations of the similarly influenced image quality attribute estimators (Secs. 5.2.2, 5.2.3, 5.3 and B.2.3). This further underpins that future studies should investigate image intensity as a third image quality attribute.

Computational Cost Analysis

We employed the FLOP/s metric in order to quantify GPU and CPU performances. Although the metric is independent of the used processing unit, it only correlates with the computational cost, but does not determine it. The actual cost further depends on the employed network architecture, processing unit architecture, processing unit clocking, software frameworks, etc. Moreover, nowadays computer systems may have other limiting factors, such as memory bandwidths of CPU and GPU processing or the I/O bandwidth when reading data from a hard disk. Therefore, it might be useful to include other performance metrics in future studies (e.g., the roofline model that further includes peak memory bandwidth [WWP09]).

We would also like to point out that the actually available CPU FLOP/s could deviate from the determined FLOP/s performances. Regarding the Intel CPU, the employed SGEMM benchmark can significantly increase the FLOP/s values when using larger matrices. For instance, we measured over 1000G FLOP/s for multiplying matrices of size $16\,\mathrm{k} \times 16\,\mathrm{k}$. Regarding the Arm CPU, there are indicators that the true performance might be higher or lower than the estimated one:

(*i*) The performance is calculated on the basis of theoretical peak performances but the CPU must at least handle OS processes concurrently.

(*ii*) Only physical CPU cores and threads are taken into account and no performance enhancing techniques such as simultaneous multithreading [TEL95].

(*iii*) The Intel and the Arm CPUs have different processor architectures, so the processing times for the same operations could vary and therefore reduce the comparability.

(*iv*) Higher FLOP/s values per clock cycle are given on the Internet (16 FLOP/s/cycle) [Use23], but without a trustworthy source.

It is also worth noting that we refrained from analyzing the computational cost in terms of a variable input size using the well-known Bachmann–Landau notation [Knu76], since we rely on standard CNN models with fixed input sizes. For more information on CNN arithmetic, the interested reader is referred to [DV16].

7.6 Summary

This chapter is dedicated to demonstrate the practical applicability of the integrated camera self-health-maintenance framework.

We first carried out an offline sensitivity analysis to determine object detection performances as a function of blur and noise image quality attributes for a transportation scenario (Sec. 7.1). Specifically, we analyzed the Udacity dataset on cars and pedestrians using the YOLOv4 and Faster R-CNN object detectors, focusing on linear motion blur and sensor noise. As a result, we obtained profiles for isolated and combined blur and noise corruptions. Especially the performance profiles for YOLOv4 demonstrated that the relation between image quality attributes and detection accuracies might be non-linear and non-trivial for ML-based methods.

In Sec. 7.2, we proposed a synthetic (*Sim*) and a real-world (*Parking Lot*) evaluation dataset. The simulation environment belonging to *Sim* allowed to emulate the application of the performance curves and the camera exposure time and ISO gain parameter changes. In the real-world *Parking Lot* scenario, we applied our framework to a real-world camera system on a robot that we navigated through a parking lot with a high exposure time to induce motion blur.

Section 7.3 provided the evaluation results. In *Sim*, the framework reduced noise and increased motion blur to improve the average car-detection rate from $\bar{p} = 0.26$ to $\bar{p} = 0.49$. In *Parking Lot*, we compared (*i*) the camera's default parameter control to (*ii*) fixed parameters and to (*iii*) our proposed control. The default control (*i*) resulted in a AP car-detection score of 26.08%, the fixed parameters (*ii*) to an AP score of 47.54%, and our control to 60.56%. In contrast to the default control that increased exposure time and thus motion blur, our framework decided to reduce exposure time at the expense of an increased ISO gain (and thus increased noise).

Subsequently, we analyzed the computational cost of our framework on real-world stationary hardware and analytically for mobile hardware (Sec. 7.4). On both platforms, the framework could be executed in real-time (19.8 and 6.6 frames per second, respectively) with the given working memory (19.8 GB for GPU and 1.8 GB for CPU operations). However, the framework's performance was limited by data pre- and post-processing running on the CPU. Potential improvements would be to vectorize CPU processings or to learn them as part of the CNN to run on the GPU, to employ the latest object detector YOLOv8, and to reduce the used data types to 16-bit floats or 8-bit integers.

Lastly, details and shortcomings of this chapter were briefly addressed in Sec. 7.5. We discussed on the resolution limits of the performance curves, on image intensity as a future research target of great importance, on computation cost metrics besides FLOP/s that incorporate more potential performance bottlenecks, and on limitations for our mobile CPU performance analysis.

CHAPTER 8

Conclusion

This chapter ends this thesis with a summary of the main findings for each research question, including limitations and a concise conclusion. Finally, a brief outlook on possible improvements and extensions of the framework is provided.

8.1 Summary

We designed, implemented and evaluated a general self-health-maintenance framework for camera systems. The primary purpose of its design was the application on autonomous mobile machines and thus to consider reliable and robust real-time operation with limited resources. To this end, the framework was developed upon novel online camera state estimators combining data-driven and physical-based models (Sensor AI), and offline-trained application performance profiles. These profiles link camera configurations to the performances of an arbitrary high-level image application allowing them to be optimized online in response to changing environments. The system was finally demonstrated on a real mobile machine for object detection and blur/noise effects caused by the camera. The development of a theoretical basis, the real-time capable implementation, and the evaluation of each framework component were guided by five research questions.

The first overarching research question focused on the framework design to meet its requirements (Ch. 4). The framework's core consists of a realistic imaging pipeline with physical models, covering a wide range of camera systems, and two sub-modules: a condition monitoring and a decision & control unit. The initial condition monitoring unit incorporates existing learning-based MTF and noise level estimators, which we have fitted to the proposed physical models, and which constantly estimate the state of the images produced in real-time. On this basis, a decision & control unit was designed that controls ISO gain and exposure time of a camera according to camera physics to manipulate blur and noise effects in a way that is found to be optimal for object detection. Altogether, this efficient modular design favors explainability, generality, physical consistency, and testing of multiple components with less effort than end-to end approaches.

The second research question addressed the extension of noise estimation to identify and quantify the sources of camera noise (Ch. 4). As a result, we developed a learning-based noise source estimator that inputs camera metadata alongside a produced image to distinguish four time-varying noise components: photon shot noise, dark current shot noise, readout noise, and residual noise. The residual noise thereby quantifies noise discrepancies between image and metadata, and is thus able to robustify the noise source estimations and to detect camera defects, miscalibrations, or misconfigurations.

The third research question aimed at evaluating accuracy and robustness of each framework component: (*i*) blur and noise estimation, (*ii*) noise source estimation, and the (*iii*) decision & control unit:

(*i*) In the blur estimation evaluation (Ch. 5), the proposed learning-based MTF estimator was found to be best suited for condition monitoring in terms of real-time requirements, arbitrary small blurs, defocus, and linear motion. Traditional blur estimators operated best for complex non-linear motion in non-real-time scenarios. In the noise estimation evaluation (Ch. 5), our learning-based approach showed best results in all scenarios. For the estimation of simultaneously occurring blur and noise, we demonstrated a simple technique to re-enabled blur estimation in the presence of strong sensor noise using an additional defocus filter. Finally, we used temporal and/or spatial aggregation of estimates to demonstrate how to effectively reduce estimator uncertainty during the online condition monitoring.

(*ii*) In the evaluation of noise source estimation (Ch. 6), only the model with access to the full set of camera metadata could accurately and robustly quantify the contribution of each noise source and even further improve total noise estimation.

(*iii*) The decision & control was successfully demonstrated in conjunction with the other components on a real camera system on a mobile robotic platform (Ch. 7). On the example of car detection, it could be shown that our framework accurately readjusted the camera parameters and significantly improved detection performance compared to fixed parameters and the built-in parameter control.

The fourth research question concerns the real-time capability of the framework on mobile hardware (Ch. 7). We found that the framework was already running in real-time on stationary and low-performance mobile hardware. The computation time and memory requirements could be further improved by using (*i*) modern mobile hardware (which we demonstrated analytically), (*ii*) vectorized or learning-based pre- and post-processings, (*iii*) the latest object detectors (e.g., YOLOv8), and (*iv*) smaller data types (e.g., 8-bit integers).

The last research question addresses the limits of the framework (Ch. 4). The applicability of the framework is primarily restricted by its underlying physical models (sensor and lens system architectures, blur and noise types). Also note that the framework targets

the optimization of only one application at a time in this initial version. It was further shown that overexposure in images negatively affects all estimators (Chs. 5 and 6). Lastly, the proposed MTF estimator requires inhomogeneous image features, which limits its fields of application.

Our research is subject to several limitations. For instance, even if Sensor AI improves robustness and explainability of learning-based approaches, they are still black boxes that should be turned more transparent in follow-up studies. Second, the learning-based methods are trained on synthetic image corruptions, which are limited in realism due to the discrepancy between simulation and reality. Finally, large-scale experiments could complete our limited real-world datasets, and the framework could be tested on modern mobile hardware to determine its current full potential.

Overall it can be concluded that with our proposed framework, camera systems can autonomously adapt to internal and external influences to optimize an envisaged image application in real-time. This thesis particularly emphasized that environmental conditions, a camera's configuration, and the intended quality of the results are inextricably linked to each other. Our ready-to-use implementation covers a wide range of camera systems and can be easily extended for many use cases of autonomous machines. However, although Sensor AI has proven not only beneficial but also necessary to determine the state of a camera, further improvements are needed to guarantee fully reliable camera systems and consequently mobile machines.

8.2 Outlook

In the course of this research work, several starting points for follow-up studies have been identified, of which the most important ones are summarized below. They all aim to improve or extend the proposed camera self-health-maintenance framework.

A straightforward improvement to the framework is to retrain the blur and noise source estimators. An improved blur estimation could account for intrinsic camera blur, which cannot be avoided and currently affects estimation accuracy. A newly trained noise source estimator could mitigate the influences of learned metadata that deviate from the noise model. In this context, the U-Net CNN architecture could be considered as well, which indicated to process an entire image faster at once. Both retrainings would further enable higher resolutions of input-output performance curves.

In addition to these direct performance improvements, there are multiple aspects of the framework that can be extended. Future studies could increase the generality by using more extended lens and noise models to cover a wider variety of camera systems. For instance, a real lens system is also temperature dependent, which can lead to material stress and thus change intrinsic parameters, and the versatility of modern camera sensor

architectures should also be represented by a corresponding noise model. Moreover, both extensions account for real-world situations that would benefit from on-board calibration during field operations to mitigate unforeseen complex effects.

The evaluations in this thesis have identified that image intensity is an important image quality attribute (for all estimators and the target application of object detection) that needs to be considered for a more reliable self-health-maintenance. Depending on the envisaged image target application, other image quality attributes such as texture or contrast could be relevant as well. However, each additional attribute increases the data dimensionality for the input-output performance curve analysis. This can be circumvented by examining for conditional independence between the attributes and by taking advantage of the high parallelizability to perform the analyses on high-performance computer clusters.

The images of a camera system are typically used for multiple applications at the same time, for which the framework can be extended as described in this thesis. On the other hand, this also increases the aforementioned data dimensionality.

Blur estimation on homogeneous image areas is another major limitation of existing estimation methods. An extension to blur source estimation (similar to the proposed noise source estimation) could mitigate the image feature requirement through the use of metadata and allow the estimation of blur sources that occur in combination.

Finally, the demonstrated use of Sensor AI techniques is only one way to counteract the black-box nature of ML-based methods, but does not solve this problem. Further studies could investigate different Sensor AI approaches that we have addressed or focus on other aspects (e.g., quantification and uncertainty reduction of learning-based methods, or decision process analysis).

APPENDIX A

Supplementary Material

A.1 Average Precision Score

Here we provide the calculation of the average precision (AP) score that complements Sec. 4.4.1 [CVV18]:

We first omit object detections B_D having an object detection confidence score $p(B_D) < 0.5$. Next, we determine whether it can be considered a match with a ground truth object detection B_{GT}, i.e., whether it can be counted as true positive (TP). This holds if

$$\exists B_{\mathrm{GT}},\ \mathrm{IOU}(B_{\mathrm{D}}, B_{\mathrm{GT}}) = \frac{B_D \cap B_{\mathrm{GT}}}{B_D \cup B_{\mathrm{GT}}} > 0.5, \tag{A.1}$$

using the so called intersection over union score $\mathrm{IOU}\colon \mathbb{R}^{2\times 2} \times \mathbb{R}^{2\times 2} \to [0,1]$[1]. We then collect the results according to their confidence scores in bins $0.1, 0.2, ..., 1.0$ and accumulate the corresponding TP values.

Subsequently, the precision and recall values on the accumulated TP can be determined using

$$\begin{aligned} \mathrm{Precision} &= \frac{\#(TP)}{N_{\mathrm{GT}}}, \\ \mathrm{Recall} &= \frac{\#(TP)}{N_{\mathrm{D}}}, \end{aligned} \tag{A.2}$$

with the number of true positives $\#(TP)$, the overall number of ground truth objects N_{GT}, and the overall number of actual object detections N_{D}. The results can be plotted as a precision-recall-curve $\mathrm{PrRe}\colon [0,1] \to [0,1]$. The final AP score is defined as the area under the precision-recall-curve

$$\mathrm{AP} = \int_0^1 \mathrm{PrRe}(x)\ dx \tag{A.3}$$

and can be computed by numerical integration.

[1] We assume that an object detection B is represented as a bounding box using two image points.

A.2 Camera System Parameters

Camera System	All. Vis. Prosilica GC1380H [Gmb21]	Ximea MQ013RG-E2 [Xim23]	Leica V-Lux (Typ 144) [Lei16]
Lens System	Cinegon 1.8/4.8-0902 [Gmb13]	Cinegon 1.8/4.8-0902	- (default)
Focal Length [mm]	4.8	4.8	9.1 – 146.0
Aperture [F-Number]	1.8	1.8	2.8 – 4.0
Sensor System	Sony ICX285	e2V EV76C661	-
Sensor Type	CCD	CMOS	CMOS
Spatial Resolution [px]	1360×1024	1280×1024	5472×3648
Pixel Pitch [µm]	6.45	5.3	2.4
Full Well Capacity [e^-]	14 000	8400	-
Dark Signal FoM* [nA/cm^2]	0.00889	0.95692	-

Table A.1: *Specifications of camera systems* used in this thesis. A hyphen denotes unknown values that are not provided by the manufacturer. $\star$: Determined in own measurements using dark frames and the method described in [Jan07, p. 171].

A.3 Real-World Noise Processing

Algorithm 1: Real-World Noise Processing

Data: Raw RN/DCSN image sessions.
Result: Offset/FPN corrected image sessions.

$s_{\text{fpn}} = 20$ ▷ `#imgs for FPN calculation`
foreach *sess* $\in$ *sessions* **do**
  if *#(sess.imgs)* $>$ s_{fpn} **then**
    $\text{imgs}_{\text{RN,fixed}} = \emptyset$
    $\text{imgs}_{\text{DCSN,fixed}} = \emptyset$
    foreach $(img_{RN}, img_{DCSN}) \in sess.imgs$ **do**
      $\mu_{\text{RN}}, \sigma_{\text{RN}} \leftarrow \text{fixNoiseDistr}(\text{img}_{\text{RN}})$
      $\text{img}_{\text{RN, fixed}} \leftarrow \text{sampleNormal}(\mu_{\text{RN}}, \sigma_{\text{RN}})$
      $\text{imgs}_{\text{RN, fixed}}.\text{insert}(\text{img}_{\text{RN, fixed}})$
      $\mu_{\text{DCSN}}, \sigma_{\text{DCSN}} \leftarrow \text{fixNoiseDistr}(\text{img}_{\text{DCSN}})$
      $\mu_{\text{DCSN}^*} = \mu_{\text{DCSN}} - \mu_{\text{RN}}$
      $\sigma_{\text{DCSN}^*} = \sqrt{\sigma^2_{\text{DCSN}} - \sigma^2_{\text{RN}}}$
      $\text{img}_{\text{DCSN, fixed}} \leftarrow \text{sampleNormal}(\mu_{\text{DCSN}^*}, \sigma_{\text{DCSN}^*})$
      $\text{imgs}_{\text{DCSN, fixed}}.\text{insert}(\text{img}_{\text{DCSN, fixed}})$
    $\text{correctFPN}(\text{imgs}_{\text{RN,fixed}}, s_{\text{fpn}})$
    $\text{correctFPN}(\text{imgs}_{\text{DCSN}^*,\text{fixed}}, s_{\text{fpn}})$

Function `fixNoiseDistr(`*img*`)`**:**
  $\text{hist} \leftarrow \text{histogram}(\text{img})$
  $x_{\max} \leftarrow \text{argmax}(\text{hist})$ **where** $x_{\max} \overset{!}{>} 0$
  $\text{hist}_{\text{fixed}} \leftarrow \text{fixHistogram}(\text{hist}, x_{\max})$
  $\mu, \sigma \leftarrow \text{fitNormal}(\text{hist}_{\text{fixed}})$
  return μ, σ

Function `fixHistogram(`*hist, x_{max}*`)`**:**
  $\text{hist}_{\text{fixed}} = \emptyset$
  foreach *(bin, val)* $\in$ *hist* **do**
    if $bin \geq 2x_{max}$ **then**
      $\text{hist}_{\text{fixed}}.\text{insert}((2x_{\max} - \text{bin}, \text{val}))$
    $\text{hist}_{\text{fixed}}.\text{insert}((\text{bin}, \text{val}))$
  return $\text{hist}_{\text{fixed}}$

Function `correctFPN(`*$imgs_{fixed}$, s_{fpn}*`)`**:**
  $\text{img}_{\text{fpn}} \leftarrow \text{meanImg}(\text{imgs}_{\text{fixed}}[0{:}s_{\text{fpn}}])$
  foreach $img_{fixed} \in imgs_{fixed}[s_{fpn}{:}]$ **do**
    $\text{img}_{\text{fixed/noFpn}} \leftarrow \text{img}_{\text{fixed}} - \text{img}_{\text{fpn}}$
    $\text{saveImg}(\text{img}_{\text{fixed/noFpn}}, \text{img.filePath})$

APPENDIX B

Supplementary Experiments

B.1 Blur Estimation

This section provides details for the blur estimation evaluations on synthetically corrupted (Sec. B.1.1) and real-world corrupted (Sec. B.1.2) datasets.

B.1.1 Synthetically corrupted Datasets

Figures. B.1, B.2, and B.3 complement the blur estimation results from Sec. 5.2.1.

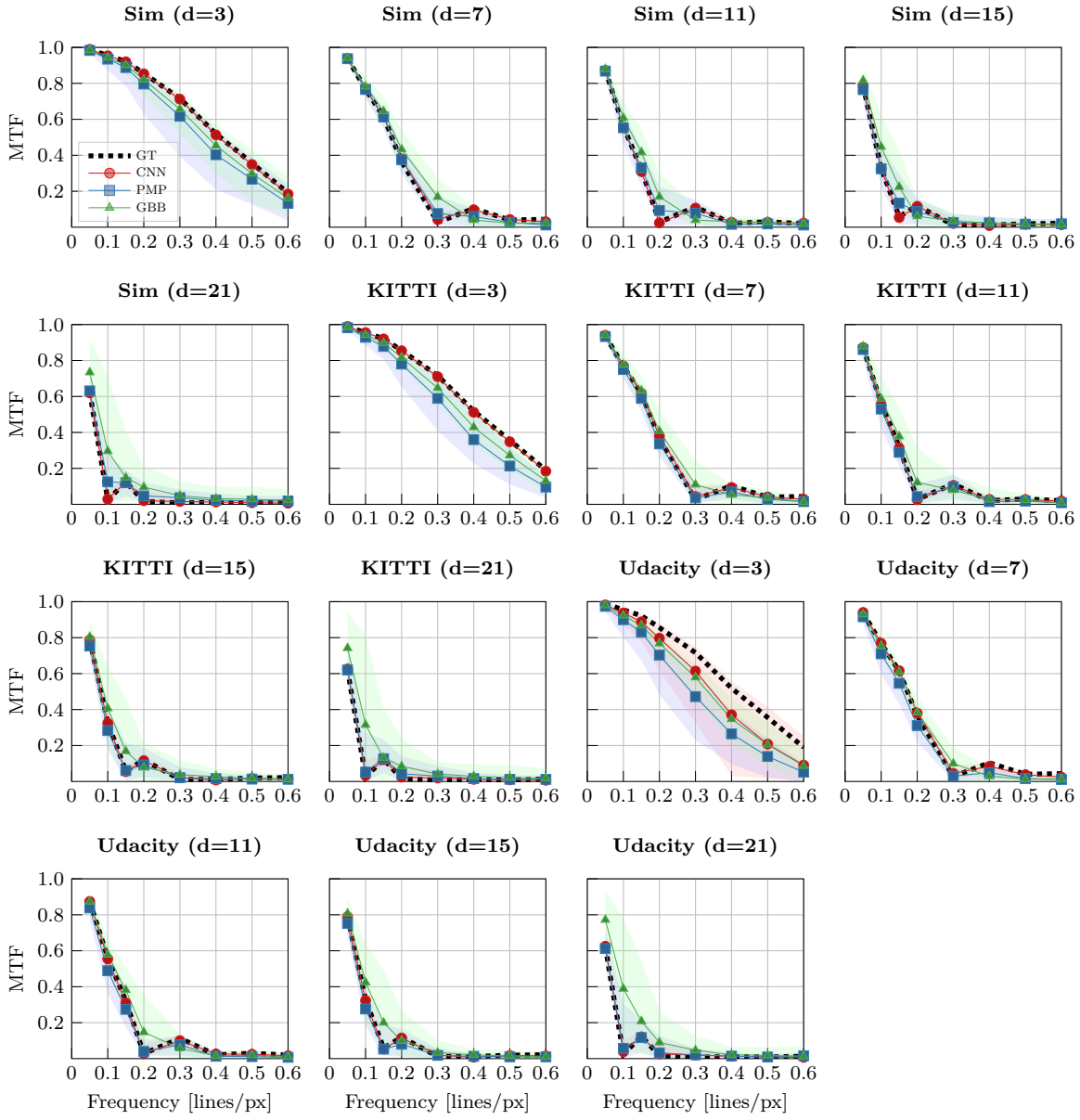


Figure B.1: *Defocus blur estimation on synthetically corrupted datasets.* Median, minimum and maximum blur estimations of the synthetically corrupted datasets (depicted by sampled points with interpolation in between and the shaded areas, respectively; vertical direction only). Details see Sec. 5.2.1.

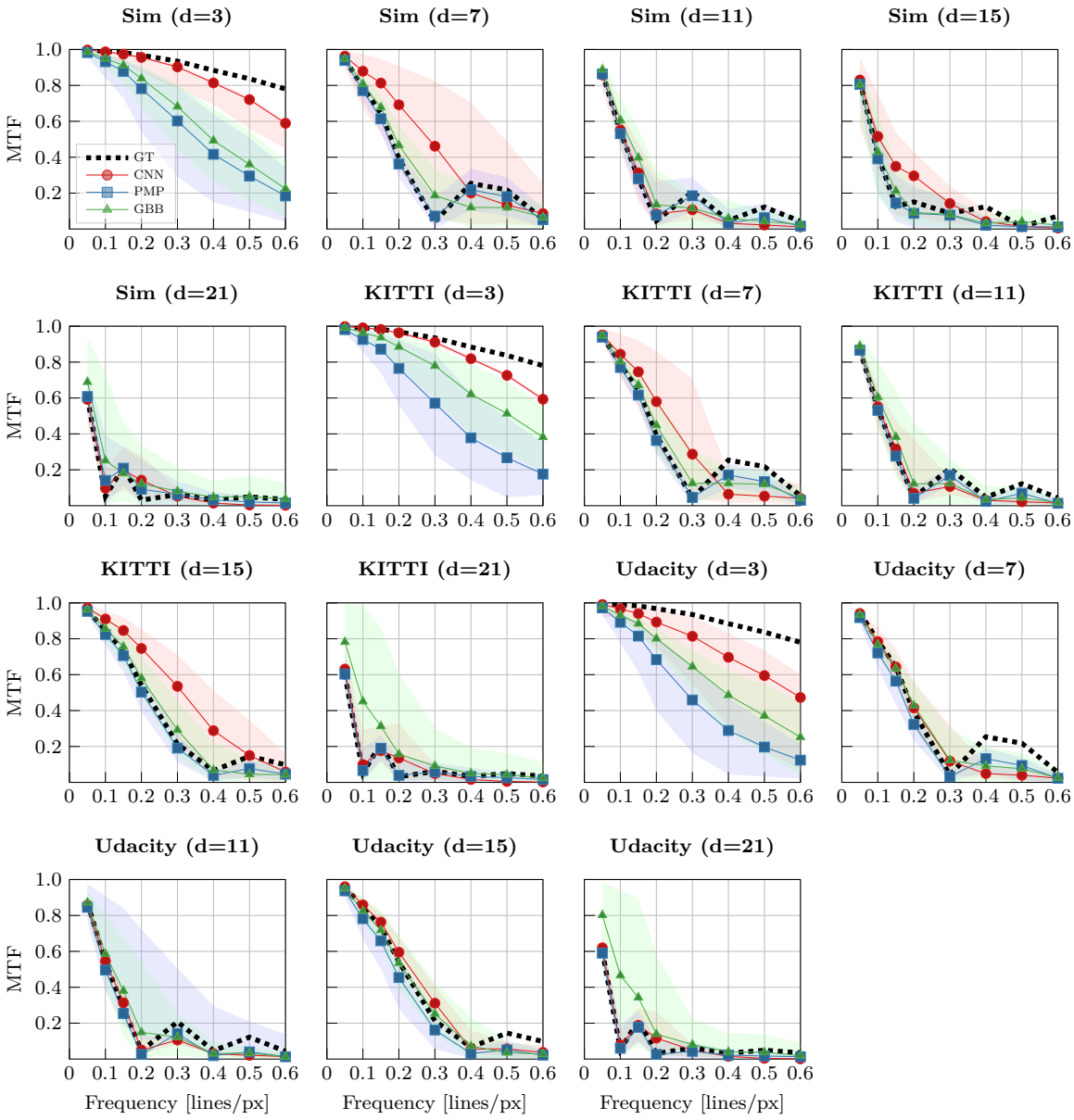


Figure B.2: *Linear motion blur estimation on synthetically corrupted datasets.* Median, minimum and maximum blur estimations of the synthetically corrupted datasets (depicted by sampled points with interpolation in between and the shaded areas, respectively; vertical direction only). Details see Sec. 5.2.1.

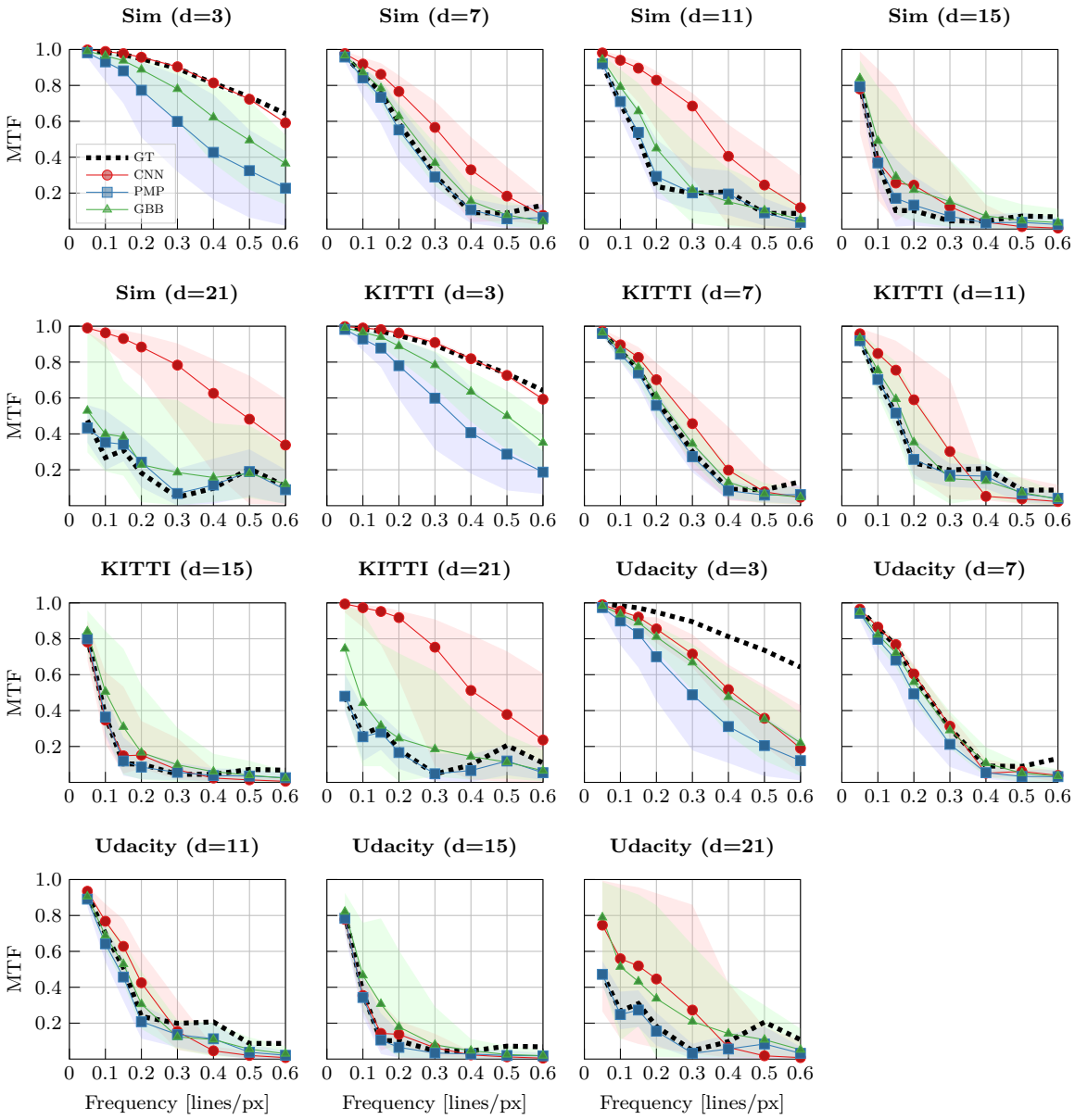


Figure B.3: *Non-linear motion blur estimation on synthetically corrupted datasets.* Median, minimum and maximum blur estimations of the synthetically corrupted datasets (depicted by sampled points with interpolation in between and the shaded areas, respectively; vertical direction only). Details see Sec. 5.2.1.

B.1.2 Real-World corrupted Datasets

The tables in Fig. B.4 contain the manually determined blur kernel sizes for the real-world DEFCARS and MOTCARS datasets specified in Sec. 5.1.2.

Real-World Defocus Blur

Object	Obj. Dist. d_O [m]	d_{PSF}[px]	$d_{\text{S.-Star}}$[px]
Car #1	6.00	19	20
	4.49	19	21
	3.36	20	21
	2.69	29	28–30
	2.23	30	32–34
	1.79	96–109	105–115
Car #2	6.00	29	30–32
	4.49	30	31
	3.36	32	31–33
	2.69	35	32–34
	2.23	36	35
	1.79	103	105
Car #3	6.00	26	26
	4.49	31	31–33
	3.36	34	35–36
	2.69	38	37–39
	2.23	52	49–51
	1.79	105–120	110–131

Real-World Motion Blur

t_{exp} [ms]	Img. ID	d_{PSF}[px]	$d_{\text{S.-Star}}$[px]
10	552	2	2
	553	14	14–16
	566	8–10	8–10
	572	14–16	15–17
	573	10–12	10–12
8	32868	14	14
	36268	14–16	14
	37368	14–16	14
	39068	18–20	20
7	163716	2	2–4
	164616	8	6–8
	165316	12	10–12
	165716	8–10	8–10
	166116	6–8	6
6	108031	14	14–16
	111731	2	2
	112331	18–19	17–18
	117431	14–16	14
	117631	14	13–15
4	219419	2	2
	219619	6	6
	219919	6–10	8–10
	220319	4–5	4
	220719	4–5	4

Figure B.4: *Manually determined real-world defocus (left) and motion (right) blur kernel sizes* using diameters of the reconstructed PSF (d_{PSF}) and the Siemens star center blur ($d_{\text{S.-Star}}$). We only keep images where both diameters differ at most 10% or 1 px (whichever is larger). The determination procedure is described in Sec. 5.1.2.

B.2 Noise Source Estimation

The following supplementary material complements the noise source estimation evaluation. It starts with details to the used camera metadata (Sec. B.2.1) and continues with extended results for the quantitative and the qualitative experiments (Secs. B.2.2 and B.2.3).

B.2.1 Camera Metadata Details

Tables B.1, B.2, and B.3 complement Tab. 4.1. Table B.1 provides the fixed camera metadata used in the adopted noise model [KW14], Tab. B.2 defines all camera parameters used in our study, and Tab. B.3 lists the specific parameter values used in the camera metadata sensitivity analysis.

B.2.2 Quantitative Experiments

Tables B.4 and B.5 extend the results from Sec. 6.2. Table B.4 provides noise estimation results on the KITTI dataset and Tab. B.5 reports real-world noise estimations for the camera *EV76C661*.

B.2.3 Experiments on Real-World Platforms

Figures B.5 to B.10 extend the results from Sec. 6.3. Figure B.5 first presents noise source estimation from the *Parking Lot* dataset recorded with the camera *ICX285* (cf. Sec. 6.3.1). Subsequently, Figs. B.6 and B.7 illustrate the noise source estimation from the *Cellar* and *Parking Lot* datasets recorded with the camera *EV76C661* (cf. Sec. 6.3.1). Figures B.8 and B.9 depict corresponding exemplary images of under- and over-exposure effects that we observed in the *Cellar* and *Parking Lot* datasets. Lastly, Fig. B.10 illustrates the noise source estimation results for the *Cellar* dataset with corrupted *sensor temperature* metadata (cf. Sec. 6.3.2).

B.2.3.1 Parking Lot

The *Parking Lot* scenario leads to similar estimations as *Cellar* (compare Fig. B.5 to Fig. 6.4). The smaller PN predictions of the noise model result from the lower scene illumination. In particular, we noticed large under-exposed areas at timestamps where $\hat{\sigma}_{\text{Noise Model}} < 2\,\text{DN}$ (Fig. B.8). Although this case is covered by the noise model and considered by our training data augmentation, it still impacts *Full-Meta* in that the method detects this model/image mismatch in the residual noise plot for the respective time stamps $\{t|[20, 22] \cup [46, 62] \cup [72, 79]\}$ s. We also observe a similar over-exposure behavior in the *Cellar* dataset (Fig. B.9). We think that these use cases are still not sufficiently represented in the training data. Note that this neither affects *w/o-Meta* (as it did not learn any residual noise) nor *Min-Meta* (which does not correctly estimate residual noise, cf. Sec. 6.3.2).

B.2.4 Experiments on Real-World Image Denoising

Table B.6 shows denoising results using the camera *EV76C661*. Details in Sec. 6.4.

Fixed Parameter	Value
Camera Offset	0 DN
CDS Gain	1
CDS s2s Time	10^{-6} s
CDS Time Factor	0.5
Flicker Noise Corner Freq.	10^{-6} Hz
Source Foll. Current Mod.	10^{-8} Hz
Source Foll. Gain	1

Table B.1: *Fixed camera metadata* in the employed noise model.

Camera Parameter	Definition
Camera Gain	Amplification factor applied to the digital camera signal.
Camera Offset	Offset value applied to the digital camera signal.
Correlated Double Sampling (CDS) Gain	Left over from CDS that is applied as gain to the signal.
CDS Sample-to-Sample Time	Time period after which a video is sampled and held within the sample-and-hold CDS circuit.
CDS Time Factor	Factor to calculate the dominant time constant from the sample-to-saple time.
Exposure Time	Time period in which the sensor is illuminated for a single image.
Flicker Noise Corner Frequency	Frequency at which the magnitude of a device's white noise and flicker noise are equal.
Sensor Temperature	Temperature of the camera sensor at image acquisition.
Dark Signal Figure of Merit	Quantity to characterize dark signal generation performance.
Full Well Capacity	Number of charge carriers a camera sensor pixel can hold.
Pixel Clock Rate	Rate at which pixels are transferred to fit an entire frame of pixels into a single refresh cycle.
Sense Node Gain	Gain applied for charge to voltage conversion.
Sense Node Reset Factor	Compensation factor of the sense node reset noise from CDS.
Sensor Pixel Size	Height/Width of a single camera sensor pixel.
Sensor Type	Construction type of the camera sensor.
Source Follower Current Modulation	Current modulation induced by burst noise.
Source Follower Gain	Voltage amplification applied by the source follower.
Thermal White Noise	White noise component within the source follower.

Table B.2: *Camera metadata definitions.*

	Uniform samples of parameter value ranges									
	Min	1	2	3	4	5	6	7	8	Max
Mean Img Intensity [DN]	0	28	56	85	113	141	170	198	226	255
Minimal Metadata										
Camera Gain*	0	8.52	12.74	15.56	17.69	19.40	20.83	22.05	23.12	24.08
Exposure Time	0.001	0.02	0.05	0.07	0.09	0.11	0.13	0.16	0.18	0.2
Sensor Temperature	0.00	8.89	17.78	26.67	35.56	44.44	53.33	62.22	71.11	80.00
Full Metadata										
Dark Signal FoM	0.00	0.11	0.22	0.33	0.44	0.55	0.66	0.77	0.88	1.00
Full Well Capacity [in k]	2.00	12.89	23.78	34.67	45.56	56.44	67.33	78.22	89.11	100.00
Pixel Clock Rate	8.00	23.78	39.56	55.33	71.11	86.89	102.67	118.44	134.22	150.00
Sense Node (SN) Gain	1.00	1.44	1.89	2.33	2.78	3.22	3.67	4.11	4.56	5.00
SN Reset Factor	0.00	0.11	0.22	0.33	0.44	0.55	0.66	0.77	0.88	1.00
Sensor Pixel Size	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	1.00
Thermal Wh. Noise**	0.10	0.76	1.41	2.07	2.72	3.38	4.03	4.69	5.34	6.00

Table B.3: *Camera metadata sensitivity analysis: sampled parameter values.* Parameter units are provided in Tab. 4.1. $\star$: Sampled from the non-dB range and converted into dB afterwards. $\star\star$: Simulated CCD sensor (CMOS sensor otherwise).

		Photon Shot Noise			DCSN			Readout Noise			Total Noise		
		Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS	Bias	Std	RMS
Random	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.18	**0.22**	**0.28**
	PGE-Net	2.03	1.16	2.33	-	-	-	-	-	-	4.00	4.80	6.24
	W/o-Meta	0.16	0.67	0.69	0.18	2.36	2.37	0.61	2.33	2.41	0.32	0.98	1.04
	Min-Meta	**0.04**	0.66	0.66	0.14	1.50	1.51	**0.04**	1.92	1.92	**0.01**	0.78	0.78
	Full-Meta	0.11	**0.14**	**0.18**	**0.05**	**0.31**	**0.32**	0.10	**0.38**	**0.40**	0.03	0.34	0.35
ICX285	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	**0.09**	0.32	0.33
	PGE-Net	2.51	1.02	2.71	-	-	-	-	-	-	3.03	1.50	3.41
	W/o-Meta	0.66	0.57	0.88	0.53	0.60	0.80	0.03	**0.64**	**0.64**	0.10	**0.03**	**0.11**
	Min-Meta	0.83	0.19	0.85	0.69	0.78	1.05	**0.02**	1.27	1.27	0.12	0.31	0.34
	Full-Meta	**0.10**	**0.12**	**0.16**	**0.16**	**0.49**	**0.51**	0.90	1.10	1.43	0.45	0.76	0.89
EV76C661	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.05	0.23	0.23
	PGE-Net	2.61	0.82	2.74	-	-	-	-	-	-	3.70	1.25	3.90
	W/o-Meta	0.60	0.52	0.79	**0.00**	1.36	1.36	**0.09**	**1.06**	**1.07**	0.12	**0.03**	**0.12**
	Min-Meta	1.11	0.24	1.14	0.43	1.18	1.26	0.76	1.49	1.67	**0.02**	0.18	0.18
	Full-Meta	**0.25**	**0.13**	**0.28**	0.78	**0.89**	**1.19**	0.43	1.73	1.78	0.21	1.03	1.05

Table B.4: *Noise source estimation on synthetically corrupted (*Random*) and real-world noised (*ICX285 *and* EV76C661*) KITTI dataset.* The best results per camera and method 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	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.14	**0.22**	**0.26**
	PGE-Net	3.02	0.95	3.17	-	-	-	-	-	-	3.65	1.13	3.82
	W/o-Meta	0.47	0.55	0.73	**0.08**	1.29	1.29	**0.32**	**1.15**	**1.19**	0.28	**0.22**	**0.26**
	Min-Meta	1.39	0.26	1.41	0.41	1.14	**1.21**	0.80	1.59	1.78	1.88	0.46	1.94
	Full-Meta	**0.28**	**0.08**	**0.30**	0.79	**0.92**	**1.21**	0.51	1.74	1.81	**0.02**	0.26	**0.26**
Tamp.17	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.31	**0.37**	0.48
	PGE-Net	3.17	1.17	3.37	-	-	-	-	-	-	3.39	1.64	3.77
	W/o-Meta	0.38	0.62	0.73	**0.09**	1.37	1.37	0.69	**1.30**	**1.47**	**0.02**	0.52	0.52
	Min-Meta	1.38	0.33	1.41	0.45	1.17	1.25	0.59	1.76	1.85	1.69	0.64	1.81
	Full-Meta	**0.34**	**0.15**	**0.37**	0.79	**0.93**	**1.22**	**0.44**	1.77	1.82	0.13	0.39	**0.41**
Udacity	$\text{DRNE}_{\text{cust.}}$	-	-	-	-	-	-	-	-	-	0.10	0.42	0.43
	PGE-Net	2.52	0.81	2.65	-	-	-	-	-	-	3.41	0.98	3.55
	W/o-Meta	0.32	0.46	0.56	**0.07**	1.32	1.32	0.48	**1.10**	**1.21**	**0.06**	0.42	0.43
	Min-Meta	0.94	0.16	0.96	0.44	1.12	1.20	0.64	1.58	1.70	1.39	0.47	1.47
	Full-Meta	**0.13**	**0.10**	**0.17**	0.72	**0.93**	**1.18**	**0.37**	1.71	1.75	0.28	**0.29**	**0.40**

Table B.5: *Noise source estimation on real-world noise extracted from the e2V EV76C661 camera 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.

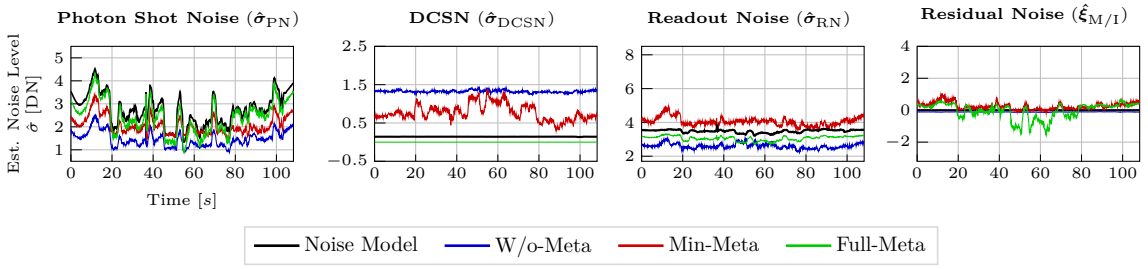


Figure B.5: *Noise source estimation (dataset:* Parking Lot*, camera:* ICX285*).* The model *Full-Meta* produces anomalies in its estimates roughly at timestamps $\{t|[20,22] \cup [46,62] \cup [72,79]\}$ s, where we noticed large under-exposed areas in the dataset (cf. Fig. B.8).

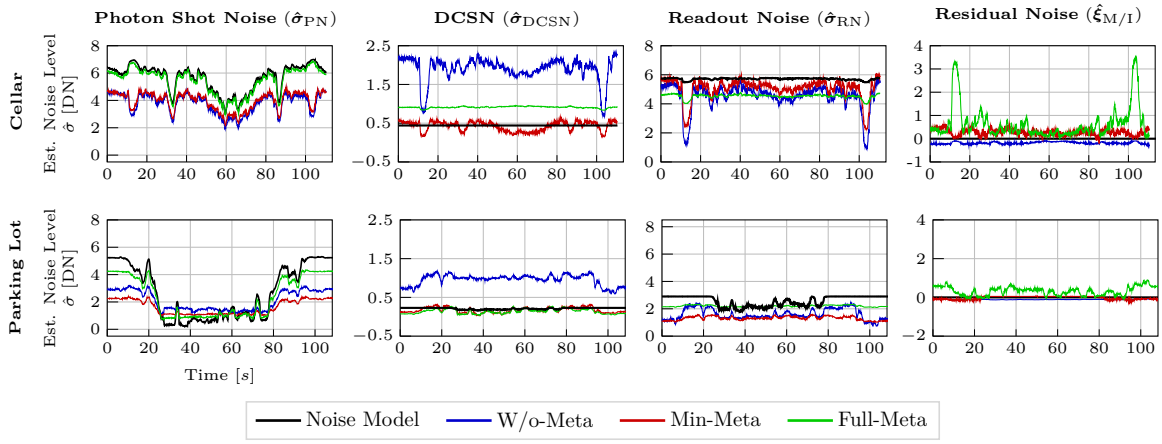


Figure B.6: *Noise source estimation (dataset:* Cellar *and* Parking Lot*, camera:* EV76C661*).* Cellar: All models produce an anomaly in their estimates roughly at timestamps $\{t|[11, 17] \cup [100, 107]\}$ s that correspond to large over-exposed areas in the images (cf. Fig. B.9).

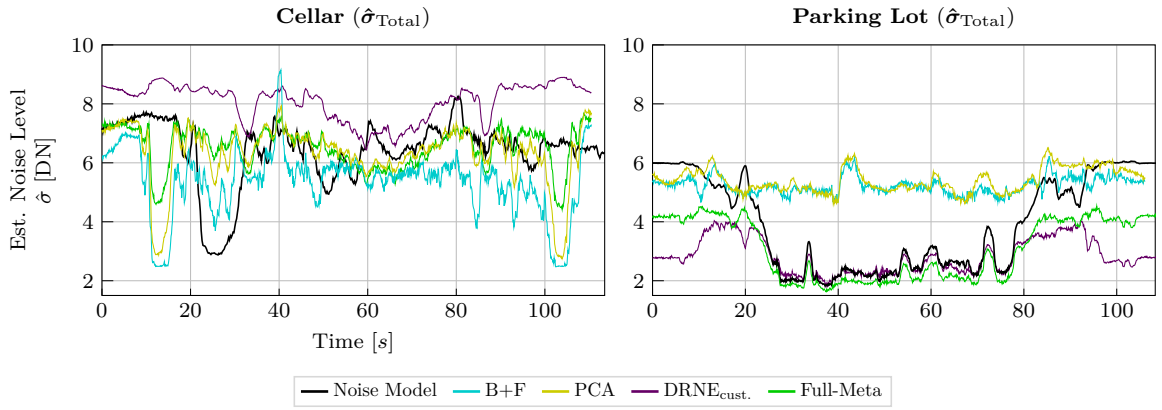


Figure B.7: *Total noise estimation (dataset:* Cellar *and* Parking Lot*, camera:* EV76C661*).* Compare to Fig. B.6. Details in Sec. 6.3.1.

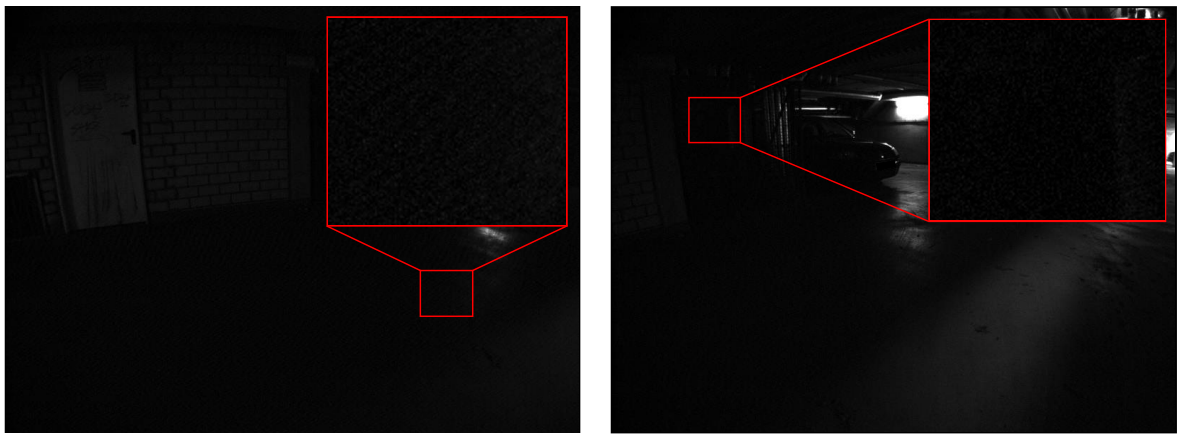


Figure B.8: *Exemplary* Parking Lot *images with under-exposed areas.* Timestamps: $t = 50$ s (Left) and $t = 60$ s (Right). Details are in Sec. 6.3.1.

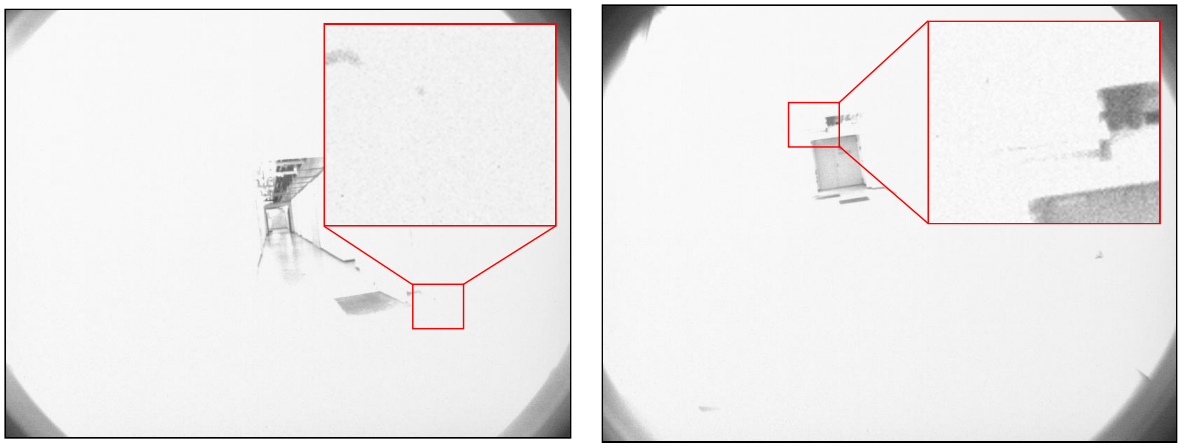


Figure B.9: *Exemplary* Cellar *images with over-exposured areas.* Timetamps: $t = 13.3\,\mathrm{s}$ (Left) and $t = 102.8\,\mathrm{s}$ (Right). Compare to Fig. B.6. Details in Sec. 6.3.1.

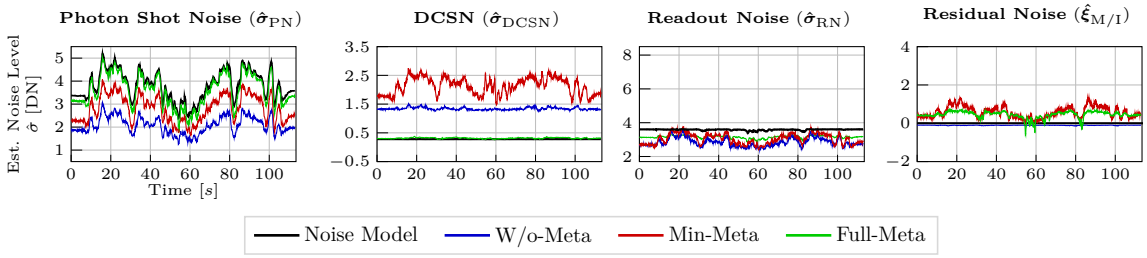


Figure B.10: *Noise source estimation on synthetically doubled* sensor temperature *metadata (dataset:* Cellar*, camera:* ICX285*).* Compare to Fig. 6.4. Details in Sec. 6.3.2.

	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	**35.36** / **0.9651**
	$\mathrm{DRNE}_{\mathrm{cust.}}$ + BM3D	33.30 / 0.9136	**33.84** / **0.9221**	34.06 / 0.9280	34.30 / 0.9342	34.85 / 0.9399
	w/o-Meta + BM3D	33.29 / 0.9132	33.83 / **0.9221**	34.08 / 0.9298	34.27 / 0.9322	34.72 / 0.9337
	Min-Meta + BM3D	33.28 / 0.9125	33.75 / 0.9192	34.05 / 0.9268	34.29 / 0.9339	34.83 / 0.9387
	Full-Meta + BM3D	**33.33** / **0.9151**	33.83 / 0.9209	**34.10** / **0.9317**	**34.37** / **0.9388**	34.83 / 0.9386
	$\mathrm{DRNE}_{\mathrm{cust.}}$ + NLM	33.22 / 0.9132	33.73 / 0.9200	33.95 / 0.9248	34.19 / 0.9296	34.71 / 0.9341
	w/o-Meta + NLM	33.22 / 0.9133	33.74 / 0.9207	33.97 / 0.9263	34.17 / 0.9283	34.62 / 0.9299
	Min-Meta + NLM	33.22 / 0.9131	33.69 / 0.9177	33.93 / 0.9238	34.18 / 0.9294	34.70 / 0.9332
	Full-Meta + NLM	33.22 / 0.9133	33.73 / 0.9219	34.00 / 0.9285	34.24 / 0.9327	34.70 / 0.9333
	FBI-Denoiser	33.18 / 0.9127	33.65 / 0.9196	33.82 / 0.9231	33.99 / 0.9251	34.42 / 0.9264
	Blind2Unblind	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	**37.78** / **0.9806**
	$\mathrm{DRNE}_{\mathrm{cust.}}$ + BM3D	35.94 / **0.9342**	36.50 / **0.9416**	37.10 / 0.9476	37.47 / 0.9537	37.33 / 0.9605
	w/o-Meta + BM3D	35.42 / 0.9198	36.38 / 0.9407	37.16 / 0.9502	37.56 / 0.9566	37.46 / 0.9661
	Min-Meta + BM3D	35.62 / 0.9262	36.11 / 0.9334	37.18 / 0.9519	37.57 / 0.9568	37.43 / 0.9649
	Full-Meta + BM3D	35.85 / 0.9324	36.43 / 0.9415	37.19 / **0.9524**	**37.82** / **0.9643**	37.70 / 0.9757
	$\mathrm{DRNE}_{\mathrm{cust.}}$ + NLM	35.67 / 0.9312	36.24 / 0.9377	36.87 / 0.9437	37.27 / 0.9498	37.78 / 0.9560
	w/o-Meta + NLM	35.52 / 0.9295	36.27 / 0.9404	36.96 / 0.9467	37.35 / 0.9522	37.30 / 0.9604
	Min-Meta + NLM	35.60 / 0.9314	36.13 / 0.9376	36.99 / 0.9483	37.36 / 0.9524	37.27 / 0.9594
	Full-Meta + NLM	35.67 / 0.9320	36.28 / 0.9403	37.00 / 0.9488	37.55 / 0.9581	37.52 / 0.9694
	FBI-Denoiser	35.58 / 0.9287	36.02 / 0.9345	36.51 / 0.9389	36.70 / 0.9420	36.59 / 0.9446
	Blind2Unblind	**36.37** / 0.8109	**36.86** / 0.8243	**37.37** / 0.8358	37.65 / 0.8446	37.51 / 0.8511

Table B.6: *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. Details see Sec. 6.4.

Bibliography

Abdelhamed, Abdelrahman, Marcus A Brubaker, and Michael S Brown (2019). "Noise flow: Noise modeling with conditional normalizing flows". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3165–3173.

Abdelhamed, Abdelrahman, Stephen Lin, and Michael S Brown (2018). "A high-quality denoising dataset for smartphone cameras". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700.

Abramov, Sergey Klavdievich et al. (2008). "Segmentation-based method for blind evaluation of noise variance in images". In: *Journal of Applied Remote Sensing* 2.023533.

Akpinar, Ugur, Erdem Sahin, and Atanas Gotchev (2019). "Learning optimal phase-coded aperture for depth of field extension". In: *IEEE International Conference on Image Processing*. IEEE, pp. 4315–4319.

Alber, Mark et al. (2019). "Integrating machine learning and multiscale modeling—perspectives, challenges, and opportunities in the biological, biomedical, and behavioral sciences". In: *NPJ digital medicine* 2.1, p. 115.

Aljadaany, Raied, Dipan K Pal, and Marios Savvides (2019). "Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10235–10244.

Aloimonos, John, Isaac Weiss, and Amit Bandyopadhyay (1988). "Active Vision". In: *International Journal of Computer Vision* 1.4, pp. 333–356.

Amer, Aishy and Eric Dubois (2005). "Fast and reliable structure-oriented video noise estimation". In: *IEEE Transactions on Circuits and Systems for Video Technology* 15.1, pp. 113–118.

Anaya, Josue and Adrian Barbu (2018). "RENOIR-A benchmark dataset for real noise reduction evaluation". In: *Journal of Visual Communication and Image Representation*.

Andersson, Olov, Fredrik Heintz, and Patrick Doherty (2015). "Model-based reinforcement learning in continuous environments using real-time constrained optimization". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 29. 1.

Arguello, Henry et al. (2023). "Deep Optical Coding Design in Computational Imaging: A data-driven framework". In: *IEEE Signal Processing Magazine* 40.2, pp. 75–88.

Arridge, Simon et al. (2019). “Solving inverse problems using data-driven models”. In: *Acta Numerica* 28, pp. 1–174.

Asim, Muhammad, Fahad Shamshad, and Ali Ahmed (2020). “Blind image deconvolution using deep generative priors”. In: *IEEE Transactions on Computational Imaging* 6, pp. 1493–1506.

Audio Video Supply Inc. (2023). URL: https://www.avsupply.com/ITM/29755/FL2-14S3C-C.html (visited on 07/03/2023).

Bacca, Jorge, Tatiana Gelvez-Barrera, and Henry Arguello (2021). “Deep coded aperture design: An end-to-end approach for computational imaging tasks”. In: *IEEE Transactions on Computational Imaging* 7, pp. 1148–1160.

Baek, Seung-Hwan et al. (2021). “Single-shot hyperspectral-depth imaging with learned diffractive optics”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2651–2660.

Bahnemiri, Sheyda Ghanbaralizadeh, Mykola Ponomarenko, and Karen Egiazarian (2022). “Learning-based noise component map estimation for image denoising”. In: *IEEE Signal Processing Letters* 29, pp. 1407–1411.

Bai, Yuanchao et al. (2018). “Graph-based blind image deblurring from a single photograph”. In: *IEEE Transactions on Image Processing* 28.3, pp. 1404–1418.

Ballard, Zachary et al. (2021). “Machine learning and computation-enabled intelligent sensor design”. In: *Nature Machine Intelligence* 3.7, pp. 556–565.

Baltrusch, Sven and Ralf Reulke (2017). “DIN-Normungsarbeit in der Photogrammetrie und Fernerkundung–Stand und Perspektiven”. In: *Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation eV* 26, pp. 280–287.

Bauer, Matthias et al. (2018). “Automatic estimation of modulation transfer functions”. In: *IEEE International Conference on Computational Photography*.

Ben-Ezra, Moshe and Shree K Nayar (2003). “Motion deblurring using hybrid imaging”. In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 1. IEEE, pp. I–I.

Bertoni, Lorenzo, Sven Kreiss, and Alexandre Alahi (2019). “Monoloco: Monocular 3d pedestrian localization and uncertainty estimation”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6861–6871.

Blakseth, Sindre Stenen et al. (2022). “Deep neural network enabled corrective source term approach to hybrid analysis and modeling”. In: *Neural Networks* 146, pp. 181–199.

Blanksby, Andrew J et al. (1997). “Noise performance of a color CMOS photogate image sensor”. In: *International Electron Devices Meeting. IEDM Technical Digest*. IEEE, pp. 205–208.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao (2020). “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv:2004.10934*.

Börner, Anko et al. (2017). “IPS–a vision aided navigation system”. In: *Advanced Optical Technologies* 6.2, pp. 121–129.

Börner, Anko et al. (2020). “Sensor Artificial Intelligence and its Application to Space Systems–A White Paper”. In: *arXiv preprint arXiv:2006.08368*.

Borodenko, Levi (2020). URL: https://github.com/LeviBorodenko/motionblur (visited on 07/03/2023).

Boyat, Ajay Kumar and Brijendra Kumar Joshi (2015). “A review paper: noise models in digital image processing”. In: *arXiv preprint arXiv:1505.03489*.

Brailean, James C et al. (1995). “Noise reduction filters for dynamic image sequences: A review”. In: *Proceedings of the IEEE* 83.9, pp. 1272–1292.

Brown, Tom et al. (2020). “Language models are few-shot learners”. In: *Advances in Neural Information Processing Systems* 33, pp. 1877–1901.

Buades, Antoni, Bartomeu Coll, and J-M Morel (2005). “A non-local algorithm for image denoising”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Vol. 2. IEEE, pp. 60–65.

Bubba, Tatiana A et al. (2019). “Learning the invisible: a hybrid deep learning-shearlet framework for limited angle computed tomography”. In: *Inverse Problems* 35.6, p. 064002.

Burningham, Norman, Zygmunt Pizlo, and Jan P Allebach (2002). “Image quality metrics”. In: *Encyclopedia of imaging science and technology* 1, pp. 598–616.

Burns, Peter D et al. (2000). “Slanted-edge MTF for digital camera and scanner analysis”. In: *Is and Ts Pics Conference*. Citeseer, pp. 135–138.

BYchao100 (2018). *Graph-Based-Blind-Image-Deblurring*. URL: https://github.com/BYchao100/Graph-Based-Blind-Image-Deblurring (visited on 07/03/2023).

Byun, Jaeseok, Sungmin Cha, and Taesup Moon (2021). “FBI-denoiser: Fast blind image denoiser for poisson-gaussian noise”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5768–5777.

Cai, Jian-Feng et al. (2011). “Framelet-based blind motion deblurring from a single image”. In: *IEEE Transactions on Image Processing* 21.2, pp. 562–572.

Carbajal, Guillermo et al. (2021). “Non-uniform blur kernel estimation via adaptive basis decomposition”. In: *arXiv preprint arXiv:2102.01026*.

Cartucho, Joao, Rodrigo Ventura, and Manuela Veloso (2018). “Robust object recognition through symbiotic deep learning in mobile robots”. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 2336–2341.

Cesarsky, CJ and A Sargent (1996). “ISOCAM in flight”. In: *Astronomy and Astrophysics* 315.2, pp. L32–L37.

Chakrabarti, Ayan (2016). “A neural approach to blind motion deblurring”. In: *IEEE European Conference on Computer Vision*, pp. 221–235.

Chan, Tony F and Chiu-Kwong Wong (1998). “Total variation blind deconvolution”. In: *IEEE Transactions on Image Processing* 7.3, pp. 370–375.

Chandrasekhar, K Vinay, Masudul H Imtiaz, and Edward Sazonov (2018). "Motion-adaptive image capture in a body-worn wearable sensor". In: *IEEE Sensors*.

Chang, Ke-Chi et al. (2020). "Learning camera-aware noise models". In: *European Conference on Computer Vision*. Springer, pp. 343–358.

Chang, Julie and Gordon Wetzstein (2019). "Deep optics for monocular depth estimation and 3d object detection". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10193–10202.

Chantas, Giannis et al. (2009). "Variational Bayesian image restoration with a product of spatially weighted total variation image priors". In: *IEEE Transactions on Image Processing* 19.2, pp. 351–362.

Chen, Guangyong, Fengyuan Zhu, and Pheng Ann Heng (2015). "An efficient statistical method for image noise level estimation". In: *IEEE International Conference on Computer Vision*.

Chen, Guangyong, Fengyuan Zhu, and Pheng Ann Heng (2015). "An Efficient Statistical Method for Image Noise Level Estimation". In: *IEEE International Conference on Computer Vision*, pp. 477–485.

Chen, Hanting et al. (2021). "Pre-trained image processing transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12299–12310.

Chen, Jingwen et al. (2018). "Image blind denoising with generative adversarial network based noise modeling". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 3155–3164.

Chen, Liang et al. (2019). "Blind image deblurring with local maximum gradient prior". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1742–1750.

Cho, Sunghyun and Seungyong Lee (2009). "Fast motion deblurring". In: *ACM SIGGRAPH Asia 2009 Papers*, pp. 1–8.

Cho, Taeg Sang et al. (2011). "Blur kernel estimation using the radon transform". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 241–248.

Chrysos, Grigorios G, Paolo Favaro, and Stefanos Zafeiriou (2019). "Motion deblurring of faces". In: *International Journal of Computer Vision* 127.6-7, pp. 801–823.

Coltman, John W (1954). "The specification of imaging properties by response to a sine wave input". In: *Journal of the Optical Society of America* 44.6, pp. 468–471.

Corner, BR, RM Narayanan, and SE Reichenbach (2003). "Noise estimation in remote sensing imagery using data masking". In: *International Journal on Remote Sensing* 24.4, pp. 689–702.

Corporation, Intel (2023). *Intel RealSense Product Family D400 Series*. URL: https://www.intelrealsense.com/download/20289/?tmstv=1680149335 (visited on 07/03/2023).

Côté, Geoffroi, Jean-François Lalonde, and Simon Thibault (2021). "Deep learning-enabled framework for automatic lens design starting point generation". In: *Optics express* 29.3, pp. 3841–3854.

— (2019a). "Extrapolating from lens design databases using deep learning." In: *Optics Express* 27.20, pp. 28279–28292.

— (2019b). "Introducing a dynamic deep neural network to infer lens design starting points". In: *Current Developments in Lens Design and Optical Engineering XX*. Vol. 11104. SPIE, pp. 8–14.

Crowell, Merton H. and Edward F. Labuda (1969). "The silicon diode array camera tube". In: *The Bell System Technical Journal* 48.5, pp. 1481–1528.

Cuomo, Salvatore et al. (2022). "Scientific machine learning through physics–informed neural networks: where we are and what's next". In: *Journal of Scientific Computing* 92.3, p. 88.

Dabov, Kostadin et al. (2007). "Image denoising by sparse 3-D transform-domain collaborative filtering". In: *IEEE Transactions on Image Processing* 16.8, pp. 2080–2095.

Davies, E.R. (2012). *Computer and Machine Vision: Theory, Algorithms, Practicalities*. Elsevier Science. ISBN: 9780123869913.

Daw, Arka et al. (2020). "Physics-guided architecture (pga) of neural networks for quantifying uncertainty in lake temperature modeling". In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, pp. 532–540.

De Stefano, Antonio, Paul R White, and William B Collis (2004). "Training methods for image noise level estimation on wavelet components". In: *EURASIP Journal on Advances in Signal Processing* 2004.16, pp. 1–8.

Deardorff, Matthew et al. (2021). "Metadata enabled contextual sensor fusion for unmanned aerial system-based explosive hazard detection". In: *Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXVI*. Vol. 11750. SPIE, pp. 91–105.

Demers, Joe (2004). "Depth of Field: A Survey of Techniques". In: *GPU Gems*. Ed. by Randima Fernando. Addison-Wesley, pp. 375–390. ISBN: 0321-228324.

Deptuch, Grzegorz et al. (2000). "Design and testing of monolithic active pixel sensors for charged particle tracking". In: *2000 IEEE Nuclear Science Symposium. Conference Record (Cat. No. 00CH37149)*. Vol. 1. IEEE, pp. 3–103.

Devore, Jay L. (2011). *Probability and Statistics for Engineering and the Sciences*. 8th. Brooks/Cole. ISBN: 9788131518397.

Doan, Nguyen Anh Khoa, Wolfgang Polifke, and Luca Magri (2019). "Physics-informed echo state networks for chaotic systems forecasting". In: *Computational Science: 19th International Conference*. Springer, pp. 192–198.

Dodge, Samuel and Lina Karam (2016). "Understanding how image quality affects deep neural networks". In: *2016 8th International Conference on Quality of Multimedia Experience*. IEEE, pp. 1–6.

Dolbeau, Romain (2015). "Theoretical Peak FLOPS per instruction set on modern Intel CPUs". In: URL: https://www.researchgate.net/publication/308804090_Theoretical_Peak_FLOPS_per_instruction_set_on_less_conventional_hardware (visited on 07/03/2023).

Dosovitskiy, Alexey et al. (2017). "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 1–16.

Dpreview (2009). *Canon unveils EOS 500D / Rebel T1i DSLR*. URL: https://www.dpreview.com/articles/6579860130/canoneos500d (visited on 07/03/2023).

Dr Robot, Inc. (2001 – 2021). *Jaguar 4x4 wheel Specification*. URL: http://jaguar.drrobot.com/specification_4x4w.asp (visited on 07/03/2023).

Dube, Brandon et al. (2017). "How good is your lens? Assessing performance with MTF full-field displays". In: *Applied Optics* 56.20, pp. 5661–5667.

Dumoulin, Vincent and Francesco Visin (2016). "A guide to convolution arithmetic for deep learning". In: *arXiv preprint arXiv:1603.07285*.

Dun, Xiong et al. (2020). "Learned rotationally symmetric diffractive achromat for full-spectrum computational imaging". In: *Optica* 7.8.

Dussault, David and Paul Hoess (2004). "Noise performance comparison of ICCD with CCD and EMCCD cameras". In: *Infrared Systems and Photoelectronic Tech.* Vol. 5563. Int. Society for Optics and Photonics.

Elder, James H and Steven W Zucker (1998). "Local scale control for edge detection and blur estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20.7, pp. 699–716.

Elmalem, Shay, Raja Giryes, and Emanuel Marom (2018). "Learned phase coded aperture for the benefit of depth of field extension." In: *Optics Express* 26.12, pp. 15316–15331.

Engeldrum, Peter G (2004). "A theory of image quality: The image quality circle". In: *Journal of imaging science and technology* 48.5, pp. 447–457.

Erichson, N Benjamin, Michael Muehlebach, and Michael W Mahoney (2019). "Physics-informed autoencoders for Lyapunov-stable fluid flow prediction". In: *arXiv preprint arXiv:1905.10866*.

Everingham, Mark et al. (2009). "The pascal visual object classes (voc) challenge". In: *International Journal of Computer Vision* 88, pp. 303–308.

Facil, Jose M et al. (2019). "CAM-Convs: Camera-aware multi-scale convolutions for single-view depth". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11826–11835.

Falanga, Davide, Kevin Kleber, and Davide Scaramuzza (2020). "Dynamic obstacle avoidance for quadrotors with event cameras". In: *Science Robotics* 5.40, eaaz9712.

Fan, Lirong and Lijun Lu (2019). "Design of a simple fisheye lens". In: *Applied Optics* 58.19, pp. 5311–5319.

Feichtner, Thorsten, Oleg Selig, and Bert Hecht (2015). "Plasmonic nanoantenna design and fabrication based on evolutionary optimization". In: *arXiv preprint arXiv:1511.05438*.

Fergus, Rob et al. (2006). "Removing camera shake from a single photograph". In: *ACM Siggraph 2006 Papers*, pp. 787–794.

Foi, Alessandro et al. (2008). "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data". In: *IEEE Transactions on Image Processing* 17.10, pp. 1737–1754.

Fossum, Eric R (1993). "Active pixel sensors: Are CCDs dinosaurs?" In: *Charge-Coupled Devices and Solid State Optical Sensors III*. Vol. 1900. International Society for Optics and Photonics, pp. 2–14.

— (2020). "The invention of CMOS image sensors: A camera in every pocket". In: *2020 Pan Pacific Microelectronics Symposium (Pan Pacific)*. IEEE, pp. 1–6.

Friedman, Milton (1975). *There's no such thing as a free lunch*. Open Court LaSalle, IL. ISBN: 9780875482972.

FWen (2019). *deblur-pmp*. URL: https://github.com/FWen/deblur-pmp (visited on 07/03/2023).

Gao, Hongyun et al. (2019). "Dynamic scene deblurring with parameter selective sharing and nested skip connections". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3848–3856.

Gawlikowski, Jakob et al. (2021). "A survey of uncertainty in deep neural networks". In: *arXiv preprint arXiv:2107.03342*.

Geiger, Andreas, Philip Lenz, and Raquel Urtasun (2012). "Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Geneva, Nicholas and Nicholas Zabaras (2020). "Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks". In: *Journal of Computational Physics* 403, p. 109056.

— (2022). "Transformers for modeling physical systems". In: *Neural Networks* 146, pp. 272–289.

Ghazal, Mohammed and Aishy Amer (2010). "Homogeneity localization using particle filters with application to noise estimation". In: *IEEE Transactions on Image Processing* 20.7, pp. 1788–1796.

GmbH, Allied Vision Technologies (2021). *Prosilica GC1380H Camera and Driver Attributes*. URL: https://cdn.alliedvision.com/fileadmin/content/documents/products/cameras/various/features/Camera_and_Driver_Attributes.pdf (visited on 07/03/2023).

GmbH, Jos. Schneider Optische Werke (2013). *3 Mega-Pixel Lens Cinegon 1.8/4.8-090*. URL: https://www.vision-dimension.com/media/pdf/7c/f2/1f/Cinegon_1-8-4-8.pdf (visited on 07/03/2023).

Goiffon, Vincent et al. (2010). "Analysis of total dose-induced dark current in CMOS image sensors from interface state and trapped charge density measurements". In: *IEEE Transactions on Nuclear Science* 57.6, pp. 3087–3094.

Gong, Dong et al. (2017). "From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2319–2328.

Gouveia, Luiz Carlos Paiva and Bhaskar Choubey (2016). "Advances on CMOS image sensors". In: *Sensor review* 36.3, pp. 231–239.

Greydanus, Samuel, Misko Dzamba, and Jason Yosinski (2019). "Hamiltonian neural networks". In: *Advances in Neural Information Processing Systems* 32.

Guo, Bingyang et al. (2020). "NERNet: Noise estimation and removal network for image denoising". In: *IEEE International Conference on Visual Communications and Image Processing* 71, p. 102851.

Guo, Shi et al. (2019). "Toward convolutional blind denoising of real photographs". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1712–1722.

Gupta, Ankit et al. (2010). "Single image deblurring using motion density functions". In: *Computer Vision: 11th European Conference on Computer Vision*. Springer, pp. 171–184.

Haim, Harel et al. (2018). "Depth estimation from a single image using deep learned phase coded mask". In: *IEEE Transactions on Computational Imaging* 4.3, pp. 298–310.

Hamamoto, Takayuki and Kiyoharu Aizawa (2001). "A computational image sensor with adaptive pixel-based integration time". In: *IEEE Journal of Solid-State Circuits* 36.4, pp. 580–585.

Han, Bin et al. (2023). "Camera Attributes Control for Visual Odometry With Motion Blur Awareness". In: *IEEE/ASME Transactions on Mechatronics (Early Access)*.

Healey, Glenn E and Raghava Kondepudy (1994). "Radiometric CCD camera calibration and noise estimation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16.3, pp. 267–276.

Hecht, E. (2017). *Optics 5th edition*. Pearson Education. ISBN: 9781292096933.

Hendrycks, Dan and Thomas Dietterich (2019). "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations". In: *International Conference on Learning Representations*.

Hershko, Eran et al. (2019). "Multicolor localization microscopy and point-spread-function engineering by deep learning". In: *Optics Express* 27.5, pp. 6158–6183.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural Computation* 9.8, pp. 1735–1780.

Holst, G.C. and T.S. Lomheim (2011). *CMOS/CCD Sensors and Camera Systems*. JCD Publishing. ISBN: 9780819486530.

Hong, Jihoon et al. (2019). "Remaining useful life prediction using time-frequency feature and multiple recurrent neural networks". In: *24th IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE, pp. 916–923.

Hradiš, Michal et al. (2015). "Convolutional neural networks for direct text deblurring". In: *Proceedings of the British Machine Vision Conference*. Vol. 10. 2.

Hu, Wei, Jianru Xue, and Nanning Zheng (2011). "PSF estimation via gradient domain correlation". In: *IEEE Transactions on Image Processing* 21.1, pp. 386–392.

Huawei Device Co., Ltd. (2023). *HUAWEI P30*. URL: https://consumer.huawei.com/de/phones/p30-backup/specs (visited on 07/03/2023).

Iglewicz, Boris and David C Hoaglin (1993). *Volume 16: how to detect and handle outliers*. Quality Press. ISBN: 9780873892605.

Igual, Jorge (2019). "Photographic Noise Performance Measures Based on RAW Files Analysis of Consumer Cameras". In: *Electronics* 8.11.

Ilardi, V. (2007). *Renaissance Vision from Spectacles to Telescopes*. American Philosophical Society. ISBN: 9780871692597.

Iliadis, Michael, Leonidas Spinoulas, and Aggelos K Katsaggelos (2020). "Deepbinarymask: Learning a binary mask for video compressive sensing". In: *Digital Signal Processing* 96, p. 102591.

Inagaki, Yasutaka et al. (2018). "Learning to capture light fields through a coded aperture camera". In: *Proceedings of the European Conference on Computer Vision*, pp. 418–434.

Irmisch, Patrick et al. (2019). "Simulation framework for a visual-inertial navigation system". In: *IEEE International Conference on Image Processing*, pp. 1995–1999.

Jahne, B. (2000). *Computer Vision and Applications: A Guide for Students and Practitioners,Concise Edition*. Elsevier Science. ISBN: 9780080502625.

Jain, Viren and Sebastian Seung (2008). "Natural image denoising with convolutional networks". In: *Advances in Neural Information Processing Systems* 21.

Janesick, J.R. (2007). *Photon Transfer*. SPIE. ISBN: 9780819467225.

— (2001). *Scientific Charge-Coupled Devices*. Vol. 83. SPIE press. ISBN: 0819436984.

Janssen, TJWM and FJJ Blommaert (1997). "Image quality semantics". In: *Journal of Imaging Science and Technology* 41.5, pp. 555–560.

Jayaraman, S., S. Esakkirajan, and T. Veerakumar (2009). *Digital Image Processing*. Tata McGraw Hill Education. ISBN: 9780070144798.

Jia, Feng et al. (2016). "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data". In: *Mechanical systems and signal processing* 72, pp. 303–315.

Jia, Xiaowei et al. (2021). "Physics-guided recurrent graph model for predicting flow and temperature in river networks". In: *Proceedings of the 2021 SIAM International Conference on Data Mining*. SIAM, pp. 612–620.

Jiang, Zhe et al. (2020). "Learning event-based motion deblurring". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3320–3329.

Jin, Kyong Hwan et al. (2017). "Deep convolutional neural network for inverse problems in imaging". In: *IEEE Transactions on Image Processing* 26.9, pp. 4509–4522.

Jocher, Glenn, Ayush Chaurasia, and Jing Qiu (2023). *YOLO by Ultralytics*. URL: https://github.com/ultralytics/ultralytics (visited on 07/03/2023).

Johnnypetr (2023). *slundberg/shap issue: multi-input CNN architecture*. URL: https://github.com/slundberg/shap/issues/559 (visited on 07/03/2023).

Joshi, Neel, Richard Szeliski, and David J Kriegman (2008). "PSF estimation using sharp edge prediction". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Joshi, Neel et al. (2010). "Image deblurring using inertial measurement sensors". In: *ACM Transactions on Graphics* 29.4, pp. 1–9.

Joung, Hyou-Arm et al. (2019). "Point-of-care serodiagnostic test for early-stage Lyme disease using a multiplexed paper-based immunoassay and machine learning". In: *ACS Nano* 14.1, pp. 229–240.

Kahana, Adar et al. (2020). "Obstacle segmentation based on the wave equation and deep learning". In: *Journal of Computational Physics* 413, p. 109458.

Kamble, Vipin Milind, Mayur Rajaram Parate, and Kishor M Bhurchandi (2019). "No reference noise estimation in digital images using random conditional selection and sampling theory". In: *The Visual Computer* 35, pp. 5–21.

Karniadakis, George Em et al. (2021). "Physics-informed machine learning". In: *Nature Reviews Physics* 3.6, pp. 422–440.

Karpatne, Anuj et al. (2017a). "Physics-guided neural networks (pgnn): An application in lake temperature modeling". In: *arXiv preprint arXiv:1710.11431* 2.

Karpatne, Anuj et al. (2017b). "Theory-guided data science: A new paradigm for scientific discovery from data". In: *IEEE Transactions on Knowledge and Data Engineering* 29.10, pp. 2318–2331.

Karumbunathan, Leela S. (2022). *NVIDIA Jetson AGX Orin Series Technbical Brief*. URL: https://www.nvidia.com/content/dam/en-zz/Solutions/gtcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf (visited on 07/03/2023).

Kashinath, Karthik et al. (2021). "Physics-informed machine learning: case studies for weather and climate modelling". In: *Philosophical Transactions of the Royal Society A* 379.2194, p. 20200093.

Kaufman, Adam and Raanan Fattal (2020). "Deblurring using analysis-synthesis networks pair". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5811–5820.

Kaufmann, R et al. (2011). "Near infrared image sensor with integrated germanium photodiodes." In: *Journal of Applied Physics* 110.2, p. 023107.

Kavadias, Spyros et al. (2000). "A logarithmic response CMOS image sensor with on-chip calibration". In: *IEEE Journal of Solid-state Circuits* 35.8, pp. 1146–1152.

Keelan, Brian (2002). *Handbook of image quality: characterization and prediction*. CRC Press. ISBN: 9780824707705.

Kettelgerdes, Marcel, Lena Böhm, and Gordon Elger (2021). "Correlating Intrinsic Parameters and Sharpness for Condition Monitoring of Automotive Imaging Sensors". In: *IEEE International Conference on System Reliability and Safety*.

Khmag, Asem et al. (2018). "Natural image noise level estimation based on local statistics for blind noise reduction". In: *The Visual Computer* 34, pp. 575–587.

Kim, Joowan, Younggun Cho, and Ayoung Kim (2018). "Generic camera attribute control using Bayesian optimization". In: *arXiv preprint arXiv:1807.10596*.

Kim, Pyojin et al. (2017). "Robust visual localization in changing lighting conditions". In: *IEEE International Conference on Robotics and Automation*. IEEE, pp. 5447–5452.

Kingslake, R. (1992). *Optics in Photography*. SPIE Optical Engineering Press. ISBN: 9780819407634.

Kirchheim, Benjamin (2009). *Testbericht: Canon EOS 500D*. URL: https://www.digitalkamera.de/Testbericht/Testbericht_Canon_EOS_500D/5781.aspx (visited on 07/03/2023).

Klette, R. (2014). *Concise Computer Vision: An Introduction into Theory and Algorithms*. Springer London. ISBN: 9781447163190.

Knuth, Donald E (1976). "Big omicron and big omega and big theta". In: *ACM Sigact News* 8.2, pp. 18–24.

Koenderink, A Femius, Andrea Alù, and Albert Polman (2015). "Nanophotonics: Shrinking light-based technology". In: *Science* 348.6234, pp. 516–521.

Konnik, Mikhail and James Welsh (2014). "High-level numerical simulations of noise in CCD and CMOS photosensors: review and tutorial". In: *arXiv preprint arXiv:1412.4031*.

Krishnan, Dilip, Terence Tay, and Rob Fergus (2011). "Blind deconvolution using a normalized sparsity measure". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 233–240.

Kühn, Stephan et al. (2020). "Analysis of package design of optic modules for automotive cameras to realize reliable image sharpness". In: *IEEE Electronics System-Integration Technology Conference*.

Kupyn, Orest et al. (2019). "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better". In: *IEEE International Conference on Computer Vision*, pp. 8878–8887.

Kupyn, Orest et al. (2018). "Deblurgan: Blind motion deblurring using conditional adversarial networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8183–8192.

Lagaris, Isaac E, Aristidis Likas, and Dimitrios I Fotiadis (1998). "Artificial neural networks for solving ordinary and partial differential equations". In: *IEEE Transactions on Neural Networks* 9.5, pp. 987–1000.

Lagaris, Isaac E, Aristidis C Likas, and Dimitris G Papageorgiou (2000). "Neural-network methods for boundary value problems with irregular boundaries". In: *IEEE Transactions on Neural Networks* 11.5, pp. 1041–1049.

Lan, Guohao et al. (2021). "MetaSense: Boosting RF sensing accuracy using dynamic metasurface antenna". In: *IEEE Internet of Things Journal* 8.18, pp. 14110–14126.

LeCun, Yann et al. (1998). "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11, pp. 2278–2324.

Lee, Dong Kyu, Junyong In, and Sangseok Lee (2015). "Standard deviation and standard error of the mean". In: *Korean Journal of Anesthesiology* 68.3, pp. 220–223.

Lee, Hyuk and In Seok Kang (1990). "Neural algorithm for solving differential equations". In: *Journal of Computational Physics* 91.1, pp. 110–131.

Lee, Minah, Burhan Ahmad Mudassar, and Saibal Mukhopadhyay (2021). "Adaptive Camera Platform Using Deep Learning-Based Early Warning of Task Failures". In: *IEEE Sensors Journal* 21.12, pp. 13794–13804.

Lehtinen, Jaakko et al. (2018). "Noise2Noise: Learning Image Restoration without Clean Data". In: *International Conference on Machine Learning*. PMLR, pp. 2965–2974.

Leica Camera AG (2016). *LEICA V-LUX (TYP 114) Technical Data*. URL: https://www.leica-camera.cn/sites/default/files/documents/2016-05/Technical-data-Leica-V-Lux%2B%28Typ%2B114%29_EN%20%281%29.pdf?fdl=1 (visited on 07/03/2023).

Levin, Anat et al. (2011). "Efficient marginal likelihood optimization in blind deconvolution". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 2657–2664.

Li, Feng, Jingyi Yu, and Jinxiang Chai (2008). "A hybrid camera for motion deblurring and depth map super-resolution". In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Li, Lerenhan et al. (2018). "Learning a discriminative prior for blind image deblurring". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6616–6625.

Liang, Jingyun et al. (2021a). "Flow-based kernel prior with application to blind super-resolution". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10601–10610.

Liang, Jingyun et al. (2021b). "Swinir: Image restoration using swin transformer". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1833–1844.

Liebe, Carl Ch (1995). "Star trackers for attitude determination". In: *IEEE Aerospace and Electronic Systems Magazine* 10.6, pp. 10–16.

Limited, Arm (2021). *Arm Cortex-A78 Core Software Optimization Guide, Revision r1p2.* URL: https://documentation-service.arm.com/static/60a5413bd63d3c31550c391e/ (visited on 07/03/2023).

Ling, Julia, Andrew Kurzawski, and Jeremy Templeton (2016). "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance". In: *Journal of Fluid Mechanics* 807, pp. 155–166.

Liu, Ce et al. (2007). "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, Dianjing et al. (2018a). "Training deep neural networks for the inverse design of nanophotonic structures". In: *ACS Photonics* 5.4, pp. 1365–1369.

Liu, Guangcan, Shiyu Chang, and Yi Ma (2014). "Blind image deblurring using spectral properties of convolution operators". In: *IEEE Transactions on Image Processing* 23.12, pp. 5047–5056.

Liu, Pengju et al. (2018b). "Multi-level wavelet-CNN for image restoration". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 773–782.

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.

— (2013). "Single-image noise level estimation for blind denoising". In: *IEEE Transactions on Image Processing* 22.12, pp. 5226–5237.

Lloyd, Gareth A and Steven J Sasson (United States Patent 4131919, 26/12/1978). *Electronic still camera.*

Lu, Boyu, Jun-Cheng Chen, and Rama Chellappa (2019). "Unsupervised domain-specific deblurring via disentangled representations". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10225–10234.

Lucy, Leon B (1974). "An iterative technique for the rectification of observed distributions". In: *The Astronomical Journal* 79, p. 745.

Lundberg, Scott M and Su-In Lee (2017). "A unified approach to interpreting model predictions". In: *Advances in Neural Information Processing Systems* 30.

Lyu, Qiongshuai, Min Guo, and Zhao Pei (2020). "DeGAN: Mixed noise removal via generative adversarial networks". In: *Applied Soft Computing* 95, p. 106478.

Ma, Kede et al. (2016). "Waterloo exploration database: New challenges for image quality assessment models". In: *IEEE Transactions on Image Processing* 26.2, pp. 1004–1016.

Ma, Ningning et al. (2018). "Shufflenet v2: Practical guidelines for efficient cnn architecture design". In: *Proceedings of the European Conference on Computer Vision*, pp. 116–131.

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* 29.

Marrero, Osvaldo (2018). “Large-numbers behavior of the sample mean and the sample median: a comparative empirical study”. In: *Revista de Matemática Teoría y Aplicaciones* 25.2, pp. 169–184.

Matsushita, Yasuyuki and Stephen Lin (2007). “Radiometric calibration from noise distributions”. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

McCloskey, Scott, Kelly Muldoon, and Sharath Venkatesha (2011). “Motion invariance and custom blur from lens motion”. In: *IEEE International Conference on Computational Photography*. IEEE, pp. 1–8.

Meißner, Henry (2020). *Determination and improvement of spatial resolution obtained by optical remote sensing systems*. Humboldt Universitaet zu Berlin (Germany). DOI: http://dx.doi.org/10.18452/22348.

Meißner, Henry et al. (2020). “Survey Accuracy and Spatial Resolution Benchmark of a camera system mounted on a fast flying drone”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*.

Mengu, Deniz et al. (2022). “At the intersection of optics and deep learning: statistical inference, computing, and inverse design”. In: *Advances in Optics and Photonics* 14.2, pp. 209–290.

Michaelis, Claudio et al. (2019). “Benchmarking robustness in object detection: Autonomous driving when winter is coming”. In: *arXiv preprint arXiv:1907.07484*.

Mohan, S, T Raghavendiran, and R Rajavel (2019). “Patch based fast noise level estimation using DCT and standard deviation”. In: *Cluster Computing* 22, pp. 14495–14504.

Moseley, Ben et al. (2021). “Extreme low-light environment-driven image denoising over permanently shadowed lunar regions with a physical noise model”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6317–6327.

Mudassar, Burhan Ahmad et al. (2019). “Camel: An adaptive camera with embedded machine learning-based sensor parameter control”. In: *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* 9.3, pp. 498–508.

Mukherjee, Mandovi et al. (2021). “Energy Efficient Pixel-Parallel Read-Out Circuits for Digital Image Sensors Using Cross-Layer Pixel Depth Control”. In: *IEEE Sensors Journal* 22.12, pp. 11317–11327.

Muralidhar, Nikhil et al. (2020). “Phynet: Physics guided neural networks for particle drag force prediction in assembly”. In: *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, pp. 559–567.

Murino, Vittorio, Gian Luca Foresti, and Carlo S Regazzoni (1996). “Adaptive camera regulation for investigation of real scenes”. In: *IEEE Transactions on Industrial Electronics* 43.5, pp. 588–600.

Naderi, Firouz and Alexander A Sawchuk (1978). "Estimation of images degraded by film-grain noise". In: *Applied Optics* 17.8, pp. 1228–1237.

Nah, Seungjun, Tae Hyun Kim, and Kyoung Mu Lee (2017). "Deep multi-scale convolutional neural network for dynamic scene deblurring". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3883–3891.

Nicholson, Angela and Alex Summersby (2023). *Image Stabilisation*. URL: https://www.canon.sk/pro/infobank/image-stabilisation-lenses (visited on 07/03/2023).

Nikon Inc. (2023). URL: https://www.nikonusa.com/en/nikon-products/product-archive/dslr-cameras/d300s.html (visited on 07/03/2023).

Nimisha, Thekke Madam, Akash Kumar Singh, and Ambasamudram N Rajagopalan (2017). "Blur-invariant deep learning for blind-deblurring". In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4752–4760.

Noroozi, Mehdi, Paramanand Chandramouli, and Paolo Favaro (2017). "Motion deblurring in the wild". In: *Pattern Recognition: 39th German Conference*. Springer, pp. 65–77.

Nuechterlein, Keith H, Raja Parasuraman, and Qiyuan Jiang (1983). "Visual sustained attention: Image degradation produces rapid sensitivity decrement over time". In: *Science* 220.4594, pp. 327–329.

Oktay, Tugrul, Harun Celik, and Ilke Turkmen (2018). "Constrained control of helicopter vibration to reduce motion blur". In: *Aircraft Engineering and Aerospace Technology*.

Oliveira, Luke de, Michela Paganini, and Benjamin Nachman (2017). "Learning particle physics by example: location-aware generative adversarial networks for physics synthesis". In: *Computing and Software for Big Science* 1.1, p. 4.

Onzon, Emmanuel, Fahim Mannan, and Felix Heide (2021). "Neural Auto-Exposure for High-Dynamic Range Object Detection". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7710–7720.

Pan, Jinshan et al. (2017a). "Deblurring images via dark channel prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10, pp. 2315–2328.

— (2017b). "Deblurring images via dark channel prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10, pp. 2315–2328.

Pan, Liyuan et al. (2019). "Phase-only image based kernel estimation for single image blind deblurring". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6034–6043.

Pankove, J.I. (1975). *Optical Processes in Semiconductors*. Dover. ISBN: 9780486602752.

Parico, Addie Ira Borja and Tofael Ahamed (2021). "Real time pear fruit detection and counting using YOLOv4 models and deep SORT". In: *Sensors* 21.14, p. 4803.

Parish, Eric J and Karthik Duraisamy (2016). "A paradigm for data-driven predictive modeling using field inversion and machine learning". In: *Journal of Computational Physics* 305, pp. 758–774.

Pedregosa, Fabian et al. (2022). *Memory Profiler*. URL: https://github.com/pythonprofilers/memory_profiler (visited on 07/03/2023).

Peter, G Engeldrum (2000). *Psychometric scaling, A toolkit for imaging system development*. Imcotek press. ISBN: 9780967870601.

Pfrommer, Samuel, Mathew Halm, and Michael Posa (2021). "Contactnets: Learning discontinuous contact dynamics with smooth, implicit representations". In: *Conference on Robot Learning*. PMLR, pp. 2279–2291.

Phillips, Jonathan B. and Henrik Eliasson (2018). *Camera Image Quality Benchmarking*. Wiley Publishing. ISBN: 1119054494.

Pimpalkhute, Varad A et al. (2021). "Digital image noise estimation using DWT coefficients". In: *IEEE Transactions on Image Processing* 30, pp. 1962–1972.

Plotz, Tobias and Stefan Roth (2017). "Benchmarking denoising algorithms with real photographs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1586–1595.

Ponomarenko, Mykola et al. (2018). "Blind estimation of white Gaussian noise variance in highly textured images". In: *Electronic Imaging* 2018.13, pp. 382–1.

Purohit, Kuldeep and AN Rajagopalan (2020). "Region-adaptive dense network for efficient motion deblurring". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 07, pp. 11882–11889.

Pyatykh, Stanislav, Jürgen Hesser, and Lei Zheng (2012). "Image noise level estimation by principal component analysis". In: *IEEE Transactions on Image Processing* 22.2, pp. 687–699.

Queiroz, Polyane Mazucatto et al. (2020). "Characteristics of radiographic images acquired with CdTe, CCD and CMOS detectors in skull radiography". In: *Imaging Science in Dentistry* 50.4, p. 339.

Radford, Alec et al. (2018). "Improving language understanding by generative pre-training". In: URL: https://www.mikecaptain.com/resources/pdf/GPT-1.pdf (visited on 07/03/2023).

Radford, Alec et al. (2019). "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8, p. 9.

Rai, Rahul and Chandan K Sahu (2020). "Driven by data or derived through physics? a review of hybrid physics guided machine learning techniques with cyber-physical system (cps) focus". In: *IEEE Access* 8, pp. 71050–71073.

Raissi, Maziar, Paris Perdikaris, and George E Karniadakis (2019). "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378, pp. 686–707.

— (2017a). "Physics Informed Deep Learning (Part II): Data-driven Discovery of Nonlinear Partial Differential Equations". In: *Computing Research Repository* abs/1711.10566.

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis (2017b). "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations". In: *arXiv preprint arXiv:1711.10561*.

Raissi, Maziar et al. (2019). "Deep learning of vortex-induced vibrations". In: *Journal of Fluid Mechanics* 861, pp. 119–137.

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.

Raskar, Ramesh, Amit Agrawal, and Jack Tumblin (2006). "Coded exposure photography: motion deblurring using fluttered shutter". In: *ACM Siggraph 2006 Papers*, pp. 795–804.

Ray, S.F. (2002). *Applied Photographic Optics*. Focal Press. ISBN: 9780240515403.

Read, Jordan S et al. (2019). "Process-guided deep learning predictions of lake water temperature". In: *Water Resources Research* 55.11, pp. 9173–9190.

Rebecq, Henri et al. (2019). "High speed and high dynamic range video with an event camera". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43.6, pp. 1964–1980.

Redmon, Joseph et al. (2016). "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788.

Ren, Dongwei et al. (2020). "Neural blind deconvolution using deep priors". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3341–3350.

Ren, Shaoqing et al. (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks". In: *Advances in Neural Information Processing Systems* 28, pp. 91–99.

Ren, Wenqi et al. (2016). "Image deblurring via enhanced low-rank prior". In: *IEEE Transactions on Image Processing* 25.7, pp. 3426–3437.

Reulke, Ralf et al. (2006). "Determination and improvement of spatial resolution of the CCD-line-scanner system ADS40". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 60.2, pp. 81–90.

Reulke, Ralf et al. (2004). "Improvement of spatial resolution with staggered arrays as used in the airborne optical sensor ADS40". In: *Proceedings of the XXth ISPRS Congress*. part B.

Reway, Fabio et al. (2020). "Test method for measuring the simulation-to-reality gap of camera-based object detection algorithms for autonomous driving". In: *2020 IEEE Intelligent Vehicles Symposium*. IEEE, pp. 1249–1256.

Richards, PL (1994). "Bolometers for infrared and millimeter waves". In: *Journal of Applied Physics* 76.1, pp. 1–24.

Richardson, William Hadley (1972). "Bayesian-based iterative method of image restoration". In: *Journal of the Optical Society of America* 62.1, pp. 55–59.

Rim, Jaesung et al. (2020). “Real-world blur dataset for learning and benchmarking deblurring algorithms”. In: *IEEE European Conference on Computer Vision*, pp. 184–201.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox (2015). “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention: 18th International Conference*. Springer, pp. 234–241.

Rudin, Leonid I, Stanley Osher, and Emad Fatemi (1992). “Nonlinear total variation based noise removal algorithms”. In: *Physica D: Nonlinear Phenomena* 60.1-4, pp. 259–268.

Saha, Priyabrata, Burhan A Mudassar, and Saibal Mukhopadhyay (2018). “Adaptive control of camera modality with deep neural network-based feedback for efficient object tracking”. In: *15th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, pp. 1–6.

Saha, Priyabrata and Saibal Mukhopadhyay (2019). “Multispectral information fusion with reinforcement learning for object tracking in IoT edge devices”. In: *IEEE Sensors Journal* 20.8, pp. 4333–4344.

Samal, Kruttidipta, Marilyn Wolf, and Saibal Mukhopadhyay (2021). “Introspective closed-loop perception for energy-efficient sensors”. In: *17th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, pp. 1–8.

San, Omer and Romit Maulik (2018). “Neural network closures for nonlinear model order reduction”. In: *Advances in Computational Mathematics* 44, pp. 1717–1750.

Sankaranarayanan, Aswin C et al. (2016). “Enhanced compressive imaging using model-based acquisition: Smarter sampling by incorporating domain knowledge”. In: *IEEE Signal Processing Magazine* 33.5, pp. 81–94.

Scholz, Jonathan et al. (2014). “A physics-based model prior for object-oriented mdps”. In: *International Conference on Machine Learning*. PMLR, pp. 1089–1097.

Schuler, Christian J et al. (2015). “Learning to deblur”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.7, pp. 1439–1451.

Senouf, Ortal et al. (2019). “Self-supervised learning of inverse problem solvers in medical imaging”. In: *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data: First MICCAI Workshop*. Springer, pp. 111–119.

Seo, Suyoung (2020). “Investigation on the Applicability of Defocus Blur Variations to Depth Calculation Using Target Sheet Images Captured by a DSLR Camera”. In: *Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography* 38.2, pp. 109–121.

Shah, Shital et al. (2018). “Airsim: High-fidelity visual and physical simulation for autonomous vehicles”. In: *Field and Service Robotics: Results of the 11th International Conference*. Springer, pp. 621–635.

Shan, Qi, Jiaya Jia, and Aseem Agarwala (2008). "High-quality motion deblurring from a single image". In: *ACM Transactions on Graphics* 27.3, pp. 1–10.

Shim, Inwook et al. (2018). "Gradient-based camera exposure control for outdoor mobile platforms". In: *IEEE Transactions on Circuits and Systems for Video Technology* 29.6.

Shin, Dong-Hyuk et al. (2005). "Block-based noise estimation using Adaptive Gaussian Filtering". In: *IEEE Transactions on Consumer Electronics* 51.1, pp. 218–226.

Shin, Ukcheol et al. (2019). "Camera exposure control for robust robot vision with noise-aware image quality assessment". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 1165–1172.

Sirignano, Justin and Konstantinos Spiliopoulos (2018). "DGM: A deep learning algorithm for solving partial differential equations". In: *Journal of Computational Physics* 375, pp. 1339–1364.

Sonoda, Toshiki, Hajime Nagahara, and Rin-ichiro Taniguchi (2014). "Motion-invariant coding using a programmable aperture camera". In: *IPSJ Transactions on Computer Vision and Applications* 6, pp. 25–33.

Specht, Donald F et al. (1991). "A general regression neural network". In: *IEEE Transactions on Neural Networks* 2.6, pp. 568–576.

Standardization, International Organization for (2017). *ISO 12233:2017 Photography — Electronic still picture imaging — Resolution and spatial frequency responses*. URL: https://www.iso.org/standard/71696.html (visited on 07/03/2023).

— (2022). *ISO/IEC 15775:2022 Information technology — Office equipment — Method of specifying image reproduction of colour copying machines and multifunction devices with copying modes by printed test charts*. URL: https://www.iso.org/standard/76913.html (visited on 07/03/2023).

Stroock, D.W. (2011). *Probability Theory: An Analytic View*. Cambridge University Press. ISBN: 9781139010825.

Suin, Maitreya, Kuldeep Purohit, and AN Rajagopalan (2020). "Spatially-attentive patch-hierarchical network for adaptive motion deblurring". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3606–3615.

Sun, H. (2016). *Lens Design: A Practical Guide*. CRC Press. ISBN: 9781351722247.

Sun, Jian et al. (2015). "Learning a convolutional neural network for non-uniform motion blur removal". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 769–777.

Sun, Jiedi, Changhong Yan, and Jiangtao Wen (2017). "Intelligent bearing fault diagnosis method combining compressed data acquisition and deep learning". In: *IEEE Transactions on Instrumentation and Measurement* 67.1, pp. 185–195.

Sun, Libin et al. (2013). "Edge-based blur kernel estimation using patch priors". In: *IEEE International Conference on Computational Photography*. IEEE, pp. 1–8.

Sun, Qilin et al. (2020a). “End-to-end learned, optically coded super-resolution SPAD camera”. In: *ACM Transactions on Graphics* 39.2, pp. 1–14.

Sun, Qilin et al. (2020b). “Learning rank-1 diffractive optics for single-shot high dynamic range imaging”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1386–1396.

Sutour, Camille, Charles-Alban Deledalle, and Jean-François Aujol (2015). “Estimation of the noise level function based on a nonparametric detection of homogeneous image regions”. In: *SIAM Journal on Imaging Sciences* 8.4, pp. 2622–2661.

Tai, Yu-Wing and Stephen Lin (2012). “Motion-aware noise filtering for deblurring of noisy and blurry images”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 17–24.

Tai, Yu-Wing et al. (2008). “Image/video deblurring using a hybrid camera”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Tai, Ying et al. (2017). “Memnet: A persistent memory network for image restoration”. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4539–4547.

Tajbakhsh, Nima et al. (2016). “Convolutional neural networks for medical image analysis: Full training or fine tuning?” In: *IEEE Transactions on Medical Imaging* 35.5, pp. 1299–1312.

Tan, Hanlin (2018). *Pixel-wise-Estimation-of-Signal-Dependent-Image-Noise*. URL: https://github.com/TomHeaven/Pixel-wise-Estimation-of-Signal-Dependent-Image-Noise-using-Deep-Residual-Learning (visited on 07/03/2023).

Tan, Hanlin et al. (2019). “Pixelwise estimation of signal-dependent image noise using deep residual learning”. In: *Computational Intelligence and Neuroscience* 2019.

Tao, Xin et al. (2018). “Scale-recurrent network for deep image deblurring”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8174–8182.

Tarantola, Albert (2005). *Inverse problem theory and methods for model parameter estimation*. SIAM. ISBN: 9780898715729.

Thung, Kim-Han and Paramesran Raveendran (2009). “A survey of image quality measures”. In: *International Conference for Technical Postgraduates*. IEEE, pp. 1–4.

Tian, Hui and Abbas El Gamal (2000). “Analysis of 1/f noise in CMOS APS”. In: *Sensors and Camera Systems for Scientific, Industrial, and Digital Photography Applications*. Vol. 3965. International Society for Optics and Photonics, pp. 168–176.

Tiwari, Shamik et al. (2014). “Blur parameters identification for simultaneous defocus and motion blur”. In: *CSI Transactions on ICT* 2, pp. 11–22.

Tjoa, Erico and Cuntai Guan (2020). “A survey on explainable artificial intelligence (xai): Toward medical xai”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.11, pp. 4793–4813.

Tobin, Josh et al. (2017). "Domain randomization for transferring deep neural networks from simulation to the real world". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 23–30.

tokusumi (2020). *keras-flops*. URL: https://github.com/tokusumi/keras-flops (visited on 07/03/2023).

Tomasi, Justin et al. (2021). "Learned camera gain and exposure control for improved visual feature detection and matching". In: *IEEE Robotics and Automation Letters* 6.2, pp. 2028–2035.

Tompson, Jonathan et al. (2017). "Accelerating eulerian fluid simulation with convolutional networks". In: *International Conference on Machine Learning*. PMLR, pp. 3424–3433.

Torres, Juan and José Manuel Menéndez (2015). "Optimal camera exposure for video surveillance systems by predictive control of shutter speed, aperture, and gain". In: *Real-Time Image and Video Processing*. Vol. 9400.

Tullsen, Dean M, Susan J Eggers, and Henry M Levy (1995). "Simultaneous multithreading: Maximizing on-chip parallelism". In: *Proceedings of the 22nd Annual International Symposium on Computer Architecture*, pp. 392–403.

Tzikas, Dimitris G, Aristidis C Likas, and Nikolaos P Galatsanos (2009). "Variational Bayesian sparse kernel-based blind image deconvolution with Student's-t priors". In: *IEEE Transactions on Image Processing* 18.4, pp. 753–764.

Udacity (2016). URL: https://github.com/udacity/self-driving-car (visited on 07/03/2023).

Unni, Rohit, Kan Yao, and Yuebing Zheng (2020). "Deep convolutional mixture density network for inverse design of layered photonic structures". In: *ACS Photonics* 7.10, pp. 2703–2712.

User ID "2A02:8109:B640:D50:7144:CE31:788B:95B5" (2023). *Floating-Point Operations Per Second (FLOPS)*. URL: https://en.wikichip.org/wiki/flops (visited on 07/03/2023).

Uss, Mikhail et al. (2011). "Image informative maps for estimating noise standard deviation and texture parameters". In: *EURASIP Journal on Advances in Signal Processing* 2011, pp. 1–12.

Uss, Mykhail L et al. (2013). "Image informative maps for component-wise estimating parameters of signal-dependent noise". In: *Journal of Electronic Imaging* 22.1, pp. 013019–013019.

Valeri, Jacqueline A et al. (2020). "Sequence-to-function deep learning frameworks for engineered riboregulators". In: *Nature Communications* 11.1, p. 5058.

Van den Bergh, Frans (2019). "Robust edge-spread function construction methods to counter poor sample spacing uniformity in the slanted-edge method". In: *Journal of the Optical Society of America* 36.7, pp. 1126–1136.

Vici, Andrea et al. (2020). “Performance and reliability degradation of CMOS Image Sensors in Back-Side Illuminated configuration”. In: *IEEE Journal of the Electron Devices Society* 8, pp. 765–772.

Waltham, Nick (2013). “CCD and CMOS sensors”. In: *Observing Photons in Space: A Guide to Experimental Space Astronomy*. Springer, pp. 423–442.

Wan, Zhong Yi et al. (2018). “Data-assisted reduced-order modeling of extreme events in complex dynamical systems”. In: *PloS one* 13.5, e0197704.

Wang, Wei et al. (2019). “Enhancing low light videos by exploring high sensitivity camera noise”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4111–4119.

Wang, Xianshun et al. (2021). “Camera Parameters Aware Motion Segmentation Network with Compensated Optical Flow”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, pp. 3368–3374.

Wang, Yafeng, Qi Lu, and Beibei Ren (2023). “Wind Turbine Crack Inspection using a Quadrotor with Image Motion Blur Avoided”. In: *IEEE Robotics and Automation Letters*.

Wang, Zejin et al. (2022a). “Blind2unblind: Self-supervised image denoising with visible blind spots”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2027–2036.

Wang, Zhendong et al. (2022b). “Uformer: A general u-shaped transformer for image restoration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17683–17693.

Wang, Zhou and Alan C Bovik (2022). *Modern image quality assessment*. Springer Nature. ISBN: 9783031011108.

Wang, Zhou et al. (2004). “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4, pp. 600–612.

Wei, Kaixuan et al. (2020). “A physics-based noise formation model for extreme low-light raw denoising”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2758–2767.

Wen, Fei et al. (2020). “A simple local minimal intensity prior and an improved algorithm for blind image deblurring”. In: *IEEE Transactions on Circuits and Systems for Video Technology*.

Westerhoff, Jens, Mirko Meuter, and Anton Kummert (2015). “A generic parameter optimization workflow for camera control algorithms”. In: *IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, pp. 944–949.

Wetzstein, Gordon et al. (2020). “Inference in artificial intelligence with deep optics and photonics”. In: *Nature* 588.7836, pp. 39–47.

Whyte, Oliver et al. (2012). “Non-uniform deblurring for shaken images”. In: *International Journal of Computer Vision* 98, pp. 168–186.

Willard, Jared et al. (2022). “Integrating scientific knowledge with machine learning for engineering and environmental systems”. In: *ACM Computing Surveys* 55.4, pp. 1–37.

Williams, Don, Peter D Burns, and Larry Scarff (2009). “Imaging performance taxonomy”. In: *Image Quality and System Performance VI*. Vol. 7242. International Society for Optics and Photonics, p. 724208.

Williams, Samuel, Andrew Waterman, and David Patterson (2009). “Roofline: an insightful visual performance model for multicore architectures”. In: *Communications of the ACM* 52.4, pp. 65–76.

Wischow, Maik et al. (2023a). *Estimating the Noise Sources of a Camera System from an Image and Metadata*. Under review.

Wischow, Maik et al. (2023b). “Monitoring and Adapting the Physical State of a Camera for Autonomous Vehicles”. In: *IEEE Transactions on Intelligent Transportation Systems*. DOI: 10.1109/TITS.2023.3328811.

Woodhouse, Chris (2015). *The Astrophotography Manual: A Practical and Scientific Approach to Deep Space Imaging*. Routledge. ISBN: 9781138055360.

Wu, Jin-Long et al. (2020). “Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems”. In: *Journal of Computational Physics* 406, p. 109209.

Xie, Saining et al. (2017). “Aggregated residual transformations for deep neural networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1492–1500.

Xie, You et al. (2018). “tempoGAN: A temporally coherent, volumetric GAN for super-resolution fluid flow”. In: *ACM Transactions on Graphics* 37.4, pp. 1–15.

Ximea GmbH (2023). URL: https://www.ximea.com/en/products/cameras-filtered-by-sensor-types/mq013rg-e2 (visited on 07/03/2023).

Xu, Jun et al. (2018). “Real-world noisy image denoising: A new benchmark”. In: *arXiv:1804.02603*.

Xu, Li and Jiaya Jia (2010). “Two-phase kernel estimation for robust motion deblurring”. In: *Computer Vision: 11th European Conference on Computer Vision*. Springer, pp. 157–170.

Xu, Li, Shicheng Zheng, and Jiaya Jia (2013). “Unnatural l0 sparse representation for natural image deblurring”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1107–1114.

Xu, Xiangyu et al. (2017). “Motion blur kernel estimation via deep learning”. In: *IEEE Transactions on Image Processing* 27.1, pp. 194–205.

Yan, Ruomei and Ling Shao (2016). “Blind image blur estimation via deep learning”. In: *IEEE Transactions on Image Processing* 25.4, pp. 1910–1921.

Yan, Yanyang et al. (2017). “Image deblurring via extreme channels prior”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4003–4011.

Yang, Dong and Jian Sun (2017). "BM3D-Net: A convolutional neural network for transform-domain collaborative filtering". In: *IEEE Signal Processing Letters* 25.1, pp. 55–59.

Yang, Jingyu et al. (2015). "Estimation of signal-dependent noise level function in transform domain via a sparse recovery model". In: *IEEE Transactions on Image Processing* 24.5, pp. 1561–1572.

Yang, Jingyu et al. (2009). "Image and video denoising using adaptive dual-tree discrete wavelet packets". In: *IEEE Transactions on Circuits and Systems for Video Technology* 19.5, pp. 642–655.

Yang, Liu, Xuhui Meng, and George Em Karniadakis (2021). "B-PINNs: Bayesian physics-informed neural networks for forward and inverse PDE problems with noisy data". In: *Journal of Computational Physics* 425, p. 109913.

Yang, Liu, Dongkun Zhang, and George Em Karniadakis (2018). "Physics-informed generative adversarial networks for stochastic differential equations". In: *arXiv preprint arXiv:1811.02033*.

— (2020). "Physics-informed generative adversarial networks for stochastic differential equations". In: *SIAM Journal on Scientific Computing* 42.1, A292–A317.

Yang, Yibo and Paris Perdikaris (2019). "Adversarial uncertainty quantification in physics-informed neural networks". In: *Journal of Computational Physics* 394, pp. 136–152.

Yao, Heng (2016). "A novel image noise level function estimation approach using camera response function constraint". In: *MATEC Web of Conferences*. Vol. 42. EDP Sciences, p. 06004.

Yao, Heng et al. (2021). "Signal-Dependent Noise Estimation for a Real-Camera Model via Weight and Shape Constraints". In: *IEEE Transactions on Multimedia* 24, pp. 640–654.

Ying Hu, Shane (2017). *Tips to Measure the Performance of Matrix Multiplication Using Intel MKL*. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/a-simple-example-to-measure-the-performance-of-an-intel-mkl-function.html (visited on 07/03/2023).

Yitzhaky, Yitzhak et al. (1998). "Direct method for restoration of motion-blurred images". In: *Journal of the Optical Society of America* 15.6, pp. 1512–1519.

Yuan, Fuh-Gwo et al. (2020). "Machine learning for structural health monitoring: challenges and opportunities". In: *Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2020* 11379, p. 1137903.

Yue, Zongsheng (2019). *Noise Level Estimation for Signal Image*. URL: https://github.com/zsyOAOA/noise_est_ICCV2015 (visited on 03/07/2023).

Zafra, Dan (2023). *Best Star Trackers for Astrophotography in 2023*. URL: https://capturetheatlas.com/best-star-trackers (visited on 07/03/2023).

Zamir, Syed Waqas et al. (2021). “Multi-stage progressive image restoration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14821–14831.

Zamir, Syed Waqas et al. (2022). “Restormer: Efficient transformer for high-resolution image restoration”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5728–5739.

Zhai, Guangtao and Xiaolin Wu (2011). “Noise estimation using statistics of natural images”. In: *IEEE International Conference on Image Processing*. IEEE, pp. 1857–1860.

Zhang, Jian, Debin Zhao, and Wen Gao (2014). “Group-based sparse representation for image restoration”. In: *IEEE Transactions on Image Processing* 23.8, pp. 3336–3351.

Zhang, Jiawei et al. (2018). “Dynamic scene deblurring using spatially variant recurrent neural networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2521–2529.

Zhang, Kai, Wangmeng Zuo, and Lei Zhang (2019). “Deep plug-and-play super-resolution for arbitrary blur kernels”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1671–1681.

Zhang, Kai et al. (2017). “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising”. In: *IEEE Transactions on Image Processing* 26.7, pp. 3142–3155.

Zhang, Kaihao et al. (2020). “Deblurring by realistic blurring”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2737–2746.

Zhang, Liang, Gang Wang, and Georgios B Giannakis (2019). “Real-time power system state estimation and forecasting via deep unrolled neural networks”. In: *IEEE Transactions on Signal Processing* 67.15, pp. 4069–4077.

Zhang, Yi et al. (2021). “Rethinking noise synthesis and modeling in raw denoising”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4593–4601.

Zhang, Yide et al. (2019). “A poisson-gaussian denoising dataset with real fluorescence microscopy images”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11710–11718.

Zhang, Zhanpeng et al. (2014). “Facial landmark detection by deep multi-task learning”. In: *Computer Vision: 13th European Conference*. Springer, pp. 94–108.

Zhu, Fengyuan, Guangyong Chen, and Pheng-Ann Heng (2016). “From noise modeling to blind image denoising”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 420–429.

Zhu, Yinhao et al. (2019). “Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data”. In: *Journal of Computational Physics* 394, pp. 56–81.

Zoran, Daniel and Yair Weiss (2011). “From learning models of natural image patches to whole image restoration”. In: *2011 International Conference on Computer Vision*. IEEE, pp. 479–486.

— (2009). “Scale invariance and noise in natural images”. In: *IEEE International Conference on Computer Vision*. IEEE, pp. 2209–2216.

Zuo, Wangmeng et al. (2016). “Learning iteration-wise generalized shrinkage–thresholding operators for blind deconvolution”. In: *IEEE Transactions on Image Processing* 25.4, pp. 1751–1764.