Technische Universität München
Fakultät für Elektrotechnik und Informationstechnik
Lehrstuhl für Integrierte Systeme

Digital Motion Picture Camera Denoising

Dipl.-Ing. Univ. Nicola Tamara Seybold

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. Gerhard Rigoll
Prüfer der Dissertation:	1. apl. Prof. Dr.-Ing. Walter Stechele 2. Univ.-Prof. Dr.-Ing. Klaus Diepold

Die Dissertation wurde am 31.07.2015 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.11.2015 angenommen.

Digital Motion Picture Camera Denoising

Tamara Seybold

Acknowledgements

First of all, I would like to express my gratitude to my supervisor Prof. Walter Stechele for initiating this dissertation. His suggestion to start a doctorate brought up the idea of this project. Furthermore, I would like to thank him for his ongoing support, academic advice and the freedom to pursue my own ideas.

Additionally, I would like to thank Prof. Klaus Diepold for being the second reviewer of this dissertation and Prof. Gerhard Rigoll for heading the committee.

I would like to thank ARRI for giving me the opportunity to write this dissertation as a member of the research and development department, which provided me financial security and direct access to the expertise and equipment of the camera development department. Performing my research within ARRI allowed this dissertation to form a bridge between theoretical research and real-world challenges.

My appreciation goes to Gunter Föttinger and Joachim Holzinger who initiated and promoted this PhD position within ARRI. I would like to thank my direct supervisor Harald Brendel for welcoming me in his team, his ongoing support and confidence in my work. He was always willing to share his in-depth knowledge on camera processing and color science, which contributed significantly to this dissertation. Moreover, I would like to express my gratitude to Dr. Stefano Andriani for the fruitful discussions on signal processing and the valuable suggestions in the final stages of the preparation of this dissertation.

My sincere appreciation goes to all my colleagues who helped me in the last years. I especially would like to thank Thomas Hach, Christian Keimel, Julian Habigt, and Michael Möller. They contributed considerably to this dissertation and many of the achievements would not have been possible without their support. Other important contributors were my students, whose excellent work was an invaluable addition to this dissertation. I would like to thank the English Writing Center at Technische Universität München, my colleagues and friends, especially Till Keyling, for the helpful comments in the final preparation of this dissertation.

A special word of gratitude to my beloved Georg for his ongoing advice and support. His helpful comments on my drafts improved this dissertation considerably. Finally, I would like to thank my family for their encouragement, love and understanding.

Abstract

Denoising images and videos captured by a digital camera receives more and more attention due to the shrinking size of today's image sensors and striving for even higher resolutions. A vast amount of research has been conducted to solve the complex problem of separating noise from the true signal. The wide-spread assumption of additive white Gaussian noise (AWGN) in readily processed image data, however, has led to algorithms that fail on real camera data. This dissertation shows that it is crucial to take the characteristics of real camera data into account in order to achieve good denoising results. Real camera noise is fundamentally different from AWGN: it is spatially and chromatically correlated, signal-dependent and the noise probability distribution is not necessarily Gaussian. New solutions are needed and the three main challenges are: first, finding denoising algorithms that work efficiently and effectively on real camera data, second, auto tuning the denoising parameter such that the denoising result is optimal for each image, and third, establishing a method to test, optimize and design algorithms with respect to the human perception of visual quality. This dissertation provides an introduction to camera processing and an in-depth analysis of the camera noise characteristics in the processing pipeline of a real camera as the basis for the design of future denoising algorithms. Further, two novel methods that can efficiently reduce realistic camera noise are proposed. As the correct parameter setting is crucial for the performance of most denoising methods, a method for automatic parameter estimation is proposed. Finally, the visual appearance of noisy and denoised videos is studied, and a method to realistically test and optimize denoising algorithms is presented. This dissertation additionally shows the importance of considering the specific noise characteristics in the design of denoising methods for depth data captured with a time-of-flight camera.

Contents

List of Tables

List of Figures

1 Introduction

Capturing the visual appearance of natural scenes is ubiquitous in our life today. We capture scenes for documentary reasons, to remember, illustrate, or prove something. Humans can observe or communicate over long distances thanks to real-time capturing, e.g., in video conferencing systems. And a vast amount of today's video data is captured for entertainment: movies for cinema, television series, commercials, and more.

Digital cameras enable us to capture these scenes. Resolution, dynamic range and color reproduction of motion picture cameras have improved significantly over the last decade – resolution nowadays ranges up to 4k, which is beyond the limit of human perception in many viewing conditions. While resolution has become higher and higher, the limits of dynamic range are still behind human perception. Human vision can adapt to a range of ten orders of magnitude, from starlight to sunlight at noon, and within a scene, five orders of magnitude can be perceived. Digital cameras, even the highest quality products, still are limited by sensor saturation in the highlights and by noise in low light conditions. Capturing in low light conditions has become even more difficult with increased sensor resolution: when the physical size remains constant, a resolution increase leads to a pixel size decrease, and thus less light is trapped by the sensor – which means more noise in the image.

So, what is noise? Technically, it is defined as a random variable adding an error to the signal value. Visually, it can be described as flickering in video sequences or "graininess" in an image. When the signal level approaches the noise level, the image quality is reduced to the point where the image content is unrecognizable. Even before this point is reached, the visual quality of the images is reduced to such an extent that it becomes useless for video applications. The noise hence limits the usable dynamic range. Reducing the noise via algorithmic methods is therefore key for capturing scenes in low light conditions, a process called noise reduction, or denoising.

An ideal denoising method would remove all the noise and show only the true underlying image. Although this goal can never be completely achieved, decades of research in signal processing have brought various powerful methods to reduce noise in natural images and videos leading to impressive results. These methods provide an image that is as close as possible to the true image, i.e., an image with low error. However, low error does not necessarily correspond to an image with high visual quality, as naturalness of the image is another important attribute. Many denoising methods can reduce the noise well, but the results appear unnatural due to denoising artifacts or oversmoothing. Therefore, denoising is rarely used in high-quality movie productions.

The goal of this dissertation is to increase visual quality of video sequences. Many existing denoising algorithms are optimized to minimize error and therefore underachieve in attaining the best visual quality. We hence need methods to reduce noise while preserving the natural character of the image. If that cannot be achieved by improving the denoising method itself, denoising strength should be adapted such that the remaining noise in the image hides the estimation error. How successfully an image can be denoised is therefore closely related to the question of how well the achieved perceptual quality can be measured.

Optimal denoising for video data should fulfill two requirements: high image quality is, of course, the first. In addition, most applications require performing the processing with limited hardware resources and in real-time, or near real-time. Obvious real-time applications include television and telepresence, but in documentary or movie productions, post-processing time is also limited. Ideally, denoising is directly applied while capturing without any delay. Thus, an ideal denoising method would process the images with low computational cost, enabling real-time processing.

Denoising research has advanced significantly over the last few decades, and the proposed denoising algorithms have become more and more complex. A discussion of whether optimality has already been reached can be found in various publications, e.g., "Is Denoising Dead?" by Chatterjee et al. [28]. However, most denoising algorithms are based on unrealistic assumptions, and therefore, they are far from optimal for denoising real camera data. The following section provides a brief overview of existing methods and examines whether they satisfy the requirements presented above.

1.1 State of the art

Decades of research on image denoising generated numerous methods producing impressive results. Recently, the most successful methods are either based on fixed basis functions, as wavelets [101] and DCT [33], training-based dictionaries [29, 36, 81], or data adapted functions based on principal component analysis (PCA) [92, 128]. For an overview of existing denoising methods see [20, 67].

Buades et al. select and compare basic methods in [20]. The authors admit the difficulty of selecting a best method, as "most original methods have caused an abundant literature proposing many improvements." While they try to select the best method from every field, they avoid hybrid methods trying to preserve the "simple and genuine character of the original method." Their comparison includes Gaussian smoothing [73], which denoises based on weighted averaging of spatially neighboring pixels, anisotropic filtering [98], total variation [104] assuming that an image has a simple geometric description (smooth areas and edges), neighborhood filters such as the bilateral filter, which take into account the greyscale value to define neighboring pixels, frequency domain filtering (classical Fourier-Wiener filter and locally adaptive filtering based on a moving window), wavelet thresholding (translation variant and invariant). Finally, they introduce their newly proposed non-local-means algorithm (NLM), which takes into account the whole block around a pixel to average it with similar blocks found within the complete image. In their comparison NLM achieves the best results; it is also the newest filter and the test images contain mostly repetitive patterns that are best handled by NLM.

In the more recent and more extensive review, Lebrun et al. [67] provide an extensive overview of recent denoising techniques and compare six recent state-of-the-art methods. Although the authors state that all algorithms show “a very similar qualitative and quantitative performance”, NLM Bayes [66], the newest method of Lebrun et al., and BM3D [33], are shown to provide the best results. However, they also compare the “method noise”, the remaining noise after denoising, and show that all algorithms leave significant low-frequency noise and some artifacts in the images.

The limitations of state-of-the-art denoising methods can be attributed to one or both of the following reasons:

- despite a low error, the perceived quality is suboptimal;
- the noise model does not correspond to real camera noise.

The best available denoising methods show an impressive increase in peak signal-to-noise ratio (PSNR) on most standard datasets and reach almost optimality in terms of image fidelity [26–28, 69, 78, 117]. However, this does not mean that the visual quality is optimal [42,123]. The PSNR usually indicates a successful reduction of noise, but it does not show how unnatural and disturbing the denoising artifacts may appear to the viewer. Vansteenkiste et al. performed an experiment with 37 observers, evaluating selected denoising results, and the outcome enables us to conclude that the PSNR ranking does not correspond to the ranking of perceived quality [122]. In this experiment, blur was usually identified as a problem, which indicates that PSNR optimal denoising results are oversmoothed, and denoising strength needs to be set to lower values to attain visually appealing results.

Various more sophisticated quality metrics have been proposed [24, 35, 87, 88, 90, 99, 105, 112, 124–126, 132]. However, these metrics face the same problem as the denoising methods itself: how to model features of natural images that are important for visual quality. Hence, they cannot yet be used as a reliable measure of visual quality.

Most denoising methods are designed and evaluated based on a standard noise model – additive white Gaussian noise (AWGN). This model does not correspond to the noise in real-world image or video data taken with a digital camera, and therefore denoising based on this model leads to clearly suboptimal results on camera data.

To understand the difference, let us review the color image capture via a digital camera, which is the usual method of image capture nowadays. One pixel captures the irradiance, thus the sensor data corresponds linearly to the brightness at the pixel position. To capture color data, a color filter array (CFA) is used, which covers the pixels with a filter layer. Thus, the output of the sensor is a value that represents the amount of light for one color band at one pixel position. This sensor data cannot be displayed before the following steps are applied: the white balance, the so-called debayering, which generates a full-color image, and the color transformations, which transform the linear data to displayable monitor data adapted to the monitor gamma and color space. These steps lead to a noise characteristic that is fundamentally different from the usually assumed AWGN: the noise is signal-dependent in the raw data; through debayering, it becomes spatially and chromatically correlated; and after the nonlinear color transformations, the noise distribution no longer corresponds to a Gaussian distribution.

Different approaches can be used to tackle the problem of spatially correlated camera noise: either apply the denoising to the raw sensor data, before debayering leads to spatial correlation, or take the correlation into account. Despite the vast number of proposed denoising methods for AWGN, only few try to employ one of these strategies.

The first approach is pursued by Hirakawa et al., who propose a wavelet-based method for CFA data [51]. A similar but locally adaptive version of the wavelet based approach is proposed by Aelterman et al. [12]. Park et al. propose a simple color transformation for CFA data denoising [97]. None of these methods have been tested on real camera data. Zhang et al. propose a principle component analysis (PCA) based solution [129] and Paliy et al. propose an efficient cross-color filtering for Bayer data in [95]; both methods are evaluated quantitatively on standard test data, but small crops of real images from a low dynamic range camera are shown. Some methods for joint debayering and denoising have been proposed [25,46,52,58,63,83,86,96,131], but they unfortunately use unrealistic datasets and/or unrealistic noise models leading to suboptimal debayering results [11]. Goossens et al. in [46] mention the signal-dependent noise model, but still propose a method for signal-independent noise.

The second approach, reducing the spatially correlated noise in the readily processed image, has hardly been studied. Kokaram et al. propose to account for a non-white noise using an estimation of the power spectral density of the noise [64]. An overview of sources of spatially correlated noise and simple adaptations of existing algorithms to suppress signal-independent non-white noise is provided by Aeltermann et al. [13]. These adaptations show good results on severely correlated noise, e.g., horizontal streaking noise occurring in thermal cameras. On demosaicked noise, however, low-frequency noise is left in the image and artifacts occur.

1.2 Research Problem

With the above findings, we conclude that research on denoising camera data must consider a more realistic noise model, while striving for low computational complexity as well as the best visual quality as perceived by a human observer. This imposes four main research questions: How can denoising methods be improved for real camera data? Which denoising methods can be implemented efficiently? How can the parameters of denoising methods be adjusted to achieve the best visual quality? How can denoising algorithms be evaluated and compared adequately and what are the important factors that influence human perception of noise in videos?

Reducing real camera noise at feasible computational costs

We have seen that denoising methods can be adapted to camera data based on two different strategies. First, using raw data allows us to reduce the noise before the subsequent steps change the noise characteristic. The second strategy is denoising the processed image, which requires taking the chromatically and spatially correlated noise into account.

Pursuing the first strategy, we are working on CFA data, and thus the neighboring values are not of the same color. As most denoising algorithms rely on a pixel neighborhood to estimate the signal at a location in the image, the challenge is to find algorithms working on CFA data. Additionally, the sensor data is high dynamic range linear sensor data, which must be treated differently than the usual readily processed test data. Hence, we need to adapt denoising methods to these signal properties and to find a realistic noise model in the raw data to take it into account in the denoising method.

The second strategy – applying the denoising step after processing the image – has pros and cons: we expect the existing algorithms to work well on that data, as they were designed based on readily processed images. Furthermore, denoising in the display-domain is close to the data that the viewer will see, and thus the human perception can be taken into account. However, the noise characteristic is extremely complex once the image has been completely processed, and therefore standard algorithms designed for AWGN are likely to fail. A trade-off must be found between a data representation close enough to the final image, and an image representation for which the noise characteristics can be well understood and modeled.

Independent of the two strategies, denoising algorithms should be able to operate using low computational resources. State-of-the-art algorithms have become extremely complex, so the question is, which denoising methods can be considered for stream-based image processing, and how can denoising methods be improved and adapted without increasing computational cost?

Optimizing denoising methods based on real camera data

Most denoising algorithms have one or more parameters that adjust the denoising strength. Optimizing them for PSNR is unrealistic for two reasons: first, the PSNR needs a reference image, which is not available in the real application. Second, the PSNR-optimized denoising result may not be optimal in terms of human perception.

Typically, the optimization for PSNR or other quality metrics leads to oversmoothed results. Hence, methods are required to optimize the parameters for the visually best denoising result, preferably with a simple method, and without requiring a reference.

Realistic evaluation of denoising methods

To compare and to choose from existing methods, as well as to evaluate newly proposed denoising methods, evaluation methods are needed that account for both a realistic noise model and human perception. A good evaluation method is critical for improving denoising algorithms.

Moreover, we need to select data that can be used for evaluating denoising algorithms. Methods are required to determine the visual quality of denoising results in a reliable manner similar to human perception. Most existing evaluation metrics are designed based on still images. While the research on quality metrics for video is advancing, no specific study on the perception of noise has been found. Therefore, we need to identify the basic properties of human vision when evaluating noisy image sequences.

1.3 Contribution and dissertation structure

Following the above described research questions, this dissertation contributes methods towards more realistic denoising research. Most of the material presented in this dissertation has been previously published in the following publications [1–8, 10], of which the author of this dissertation has been the main contributor. The content of this dissertation additionally presents the author's contribution to the following publications [9, 11].

This dissertation starts by providing an overview of the steps of a camera processing pipeline in Chapter 2. Subsequently, the noise characteristic after each step is analyzed based on measurements and simulations. State-of-the-art denoising methods are selected and applied to real camera data. The results are evaluated to determine which algorithms achieve high visual quality at the different steps of the processing pipeline. Additionally, small adaptations of the existing algorithms are presented that may improve denoising on real camera data.

Chapter 3 presents two new methods for denoising real camera data. First, a method is proposed for efficient raw data denoising that is based on a shape-adaptive

DCT (SA-DCT), originally proposed for non-CFA data. The method consists of three steps: a luminance transformation of the Bayer data, determining an adequate neighborhood for denoising, and hard thresholding in the SA-DCT domain. The SA-DCT is applied to realistic CFA data and accounts for the signal-dependent noise characteristic using a locally adaptive threshold and signal-dependent weights. Additionally, a computationally efficient solution to suppress flickering in video data is presented. The method is evaluated quantitatively and visually using both realistically simulated test sequences and real camera data.

The second method is a wavelet denoising method that can operate on RGB data taking into account the spatial correlation of camera noise after debayering. The wavelet transformation we use is very simple, so that it can be implemented with very low computational complexity. The noise characteristics for the different steps in the color processing are analyzed and the impact of real camera noise on the wavelet coefficients is investigated. Real camera noise measurements and simulation of the processing steps allow us to find a good approximation for the noise characteristics, which can be used subsequently in standard wavelet denoising methods. Wavelet hard thresholding and bivariate thresholding are adapted based on our noise analysis results. Both the visual quality and objective quality metrics are shown. As the method is implemented using look-up-tables that are pre-calculated, it can be implemented with very low computational complexity and can be used for processing HD video sequences in real-time on an FPGA.

To obtain optimal denoising results with respect to visual quality, the correct parameter setting is crucial. Chapter 4 therefore proposes a method that automatically estimates the optimal parameter of a denoising algorithm. The proposed approach compares the gradient histogram of a denoised image to an estimated reference gradient histogram. The reference gradient histogram is estimated based on down- and upsampling of the noisy image. Thus, the method works without a reference and is image-adaptive. For evaluating the proposed down-/upsampling-based gradient histogram method (DUG), a subjective test with 20 participants is performed, including images from both the Kodak data set and the more realistic ARRI data set. The parameters of seven denoising methods, including the state-of-the-art method BM3D, are optimized based on the proposed method. The subjective test results show that the parameter estimated by the DUG method is very close to the optimal parameter in terms of human perception. Despite being very fast and simple to implement, the method shows a lower error than all other tested no-reference metrics.

Chapter 5 focuses on the problem of realistically evaluating denoising results: based on the new camera noise model presented in Sec. 2.2, Sec. 5.1 describes how to include the full processing chain of a single sensor camera to obtain realistic test data. A subjective test with 18 participants evaluates the visual quality of noisy and denoised test sequences reliably. Results show that the noise characteristics have a significant effect on visual quality. To find out whether quality metrics reflect the test results, the performance of 12 full-reference metrics is evaluated, and – as no-reference metrics are especially useful for denoising parameter tuning – additionally five no-reference metrics are evaluated. Unfortunately, the metrics underachieve in estimating the visual quality of the test sequences. To improve the quality estimation of digital images and videos and to advance research on denoising algorithms, a more realistic noise model should be used in future research.

For a better understanding of the human perception of camera noise and the perception of denoised sequences, Sec. 5.2 investigates the visibility of noise in images and videos. A subjective test with nine observers compares static (spatial) noise with dynamic (spatiotemporal) noise. Eight different spatial frequency bands are used for each noise type. The results indicate that the sensitivity to spatially low-frequency noise is significantly higher with temporal variation. This significant increase in visibility can be included in the design of future video denoising methods and lead to better visual quality.

Chapter 6 addresses time-of-flight data denoising. Time-of-flight cameras are used to obtain depth data describing the 3D information of a scene. Mounted on a regular color camera, this depth data is captured by phase measurement simultaneously with the color data. Besides strong signal-dependent Gaussian noise, a significant amount of salt-and-pepper noise, caused by the ambiguity in the depth measurements, is visible. The chapter proposes a pre-processing algorithm that is able to isolate and correct affected pixels by employing the physical behavior of the capturing system. The salt-and-pepper noise is eliminated and Gaussian noise remains, so that standard denoising can be applied. Using this pre-processing filter before applying a denoising step clearly improves the denoising results. The significant improvement shows the importance of considering the specific noise characteristics in the design of denoising methods for time-of-flight camera data.

Finally, Chapter 7 concludes and summarizes the main findings of this dissertation.

2 Beyond Standard Noise Models: Noise and Denoising in a Real Camera Processing Pipeline

As mentioned in the introduction, research on denoising should consider a more realistic noise model to achieve high visual quality. Therefore, this chapter provides the basis for a more realistic denoising research: it describes a real camera processing pipeline, analyzes the noise characteristics in this pipeline, and provides an overview on denoising camera data based on existing denoising methods.

Understanding the signal properties and noise characteristics of a digital color camera requires to get familiar with the camera processing first. The raw sensor data cannot directly be displayed – to obtain a monitor- or display-domain image, the processing steps white balance, debayering, and color transformations are necessary. In Sec. 2.1 these steps are explained in detail.

The second part of this chapter analyzes the noise characteristics in the processing pipeline (Sec. 2.2). First, measurements of the noise in the sensor data are presented. Subsequently, the noise characteristics are analyzed after each of the processing steps. This analysis, which draws from previous publications [1,5,10], respects all steps of the camera processing that contribute to the final display-domain image. In addition to the pixel domain analysis, e.g., the spatial correlation of the noise, it extends to the frequency domain allowing to better match the results to existing models of human perception and to easier connect the results with traditional filtering theory. As film grain is often considered to be more visually appealing than today's digital camera noise, the characteristics of film grain noise are additionally analyzed and compared to the digital camera noise.

The third part of this chapter, Sec. 2.3, presents the implications of real camera noise characteristics, which differ from the usually assumed noise model, on the performance of denoising methods. The amount of denoising methods published in the last decades is uncountable, and hence, this dissertation does not claim to provide a complete overview. Instead, from simple to sophisticated and from non-local means to sparse dictionary, various methods are selected. For each method, we discuss how it can be applied to real camera data; however, the contribution is not limited to that. In most of the following sections, algorithms are adapted to allow a better performance on camera data, e.g., including the camera noise model

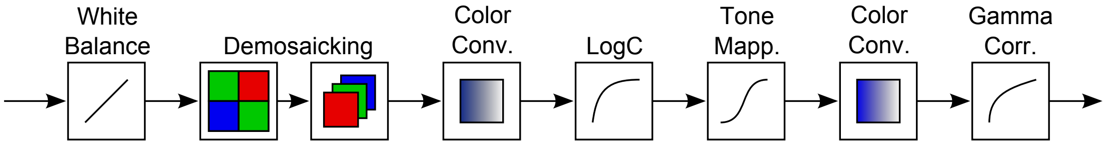


Figure 2.1: The complete processing pipeline including all steps that were described.

in Sec. 2.3.1 or adapting an algorithm proposed for grayscale data to color images in Sec. 2.3.4.

Most of the following description is based on the sensor of the ARRI ALEXA camera. The ALEXA camera has been developed for motion picture recordings in digital cinema applications. It has a CMOS sensor with a resolution of 3414x2298 pixels cropped depending on the user's needs. This dissertation refers to the 16:9 2880x1620 crop because it is the most widely used one. In front of the sensor, the camera has a filter pack composed of an infrared cut-off filter, an ultraviolet cut-off filter, and an optical low-pass filter to reduce aliasing. The color filter array (CFA) located between the filter pack and the sensor is a Bayer mask. This camera captures high-quality motion picture data and can deliver the data uncompressed in 12 bit precision (logarithmic data, corresponds to 16 bit linear data). Another advantage is that the camera processing steps are published [11]. Access to the raw data and the published processing steps provide good reasons to use the ARRI ALEXA camera, but the methods presented in this dissertation are not restricted to that camera alone; other cameras may be used and analyzed equivalently.

2.1 Camera Processing

The processing of the raw sensor data captured with a single-sensor camera needed to obtain the displayable monitor image is presented in this section. The complete processing pipeline is shown in Fig. 2.1.

The first step – white balance – applies a gain factor to each color channel correcting the white point according to the illuminant in the scene (Sec. 2.1.1).

The second step is the demosaicking, which generates a full-color image based on the sensor data, which contains only one color value per pixel. The process is also called debayering when a sensor covered with a Bayer color filter array (CFA) is used. As the Bayer pattern is the most usual CFA, this dissertation mainly studies debayering, but most of the results may be also valid for other CFA. Sec. 2.1.2 describes the debayering problem and mentions the state of the art in debayering methods with a focus of the methods used later in this dissertation.

The third step includes the color transformations. We show the basic color processing steps in Sec. 2.1.3. A minimum requirement is to adapt to the adequate monitor color space and to account for the monitor gamma. This basic processing, however, is not sufficient for today's high-dynamic-range sensor data. On such data, more sophisticated nonlinear processing can be applied to obtain a higher visual quality of the final image data. The additional nonlinear transformations are presented in Sec. 2.1.4.

2.1.1 White Balance

White balance corrects the white point according to the illuminant in the scene, such that equal RGB values represent achromatic objects.

The sensor is covered with a CFA, it captures the R-, G- and B- irradiance according to the spectral sensitivity of the photo site L_P and the transmission characteristic of the filters. Three types of filters influence the effective radiance that can be captured by the sensor. The UV filter, with its spectral transmission distribution L_{UV}, the infrared cut-off filter with its spectral transmission distribution L_{IR}, and the color filter, with a spectral transmission distribution $L_{CFA,c}$ for each channel $c \in [R, G, B]$. The result is the tristimulus value r_c.

$$r_c = \int_\lambda L_P L_{UV}\, L_{IR}\, L_{CFA,c}\, r(\lambda)\,\mathrm{d}\lambda = \int_\lambda L_c(\lambda)\, r(\lambda)\,\mathrm{d}\lambda \qquad (2.1)$$

The spectral radiance in the scene $r(\lambda)$ depends on the reflectance of objects and the light sources in the scene, and so does r_c. However, independent of the light source that illuminated the scene, achromatic objects are expected to be represented by equal RGB values in all usual RGB color spaces. Thus, as a first step, white balance must be applied accounting for the scene illumination.

To describe the illumination in a scene, the term color temperature is widely used. It describes that the light source has the same color like a black-body radiator with that absolute temperature. A black-body radiator is a theoretical illuminant emitting light according to Planck's law. Increasing color temperature means that a light source becomes bluer, or less red. The camera provides presets for the white balance according to these illuminants.

Mathematically, white balance is a simple linear process: a constant gain is applied to the three different color channels depending on the selected color temperature. In the image example in Fig. 2.2a and (b) the sensor data before and after the white balance is shown. The images do not show RGB data but raw data from the Bayer sensor, and hence the Bayer pattern is visible. As the white balance reduces color differences in white or grey areas, the pattern of the color filter array is less visible in white areas, e.g., the white parts of the traffic cone in Fig. 2.2b compared to Fig. 2.2a.

2.1.2 Debayering: from sensor data to full color RGB images

The sensor data contains only one value per pixel, which is green, red or blue depending on the location of the pixel. The most usual CFA is the Bayer pattern,

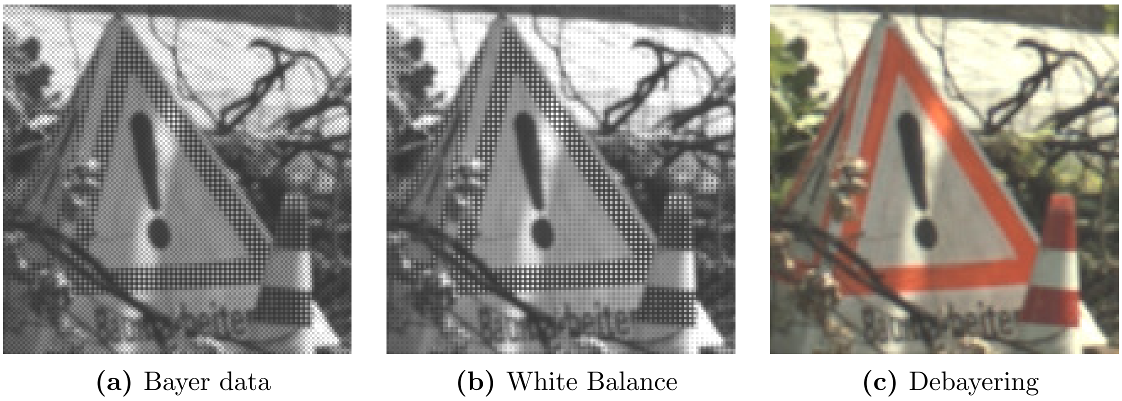


Figure 2.2: The first step in processing the Bayer data (a) is the white balance, result in (b). The debayering leads to a full-color RGB image (c).

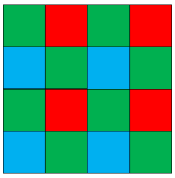


Figure 2.3: Bayer pattern

shown in Fig. 2.3. To obtain a full-color image with three color values per pixel, an interpolation method is applied, called color demosaicking or debayering. In this section we introduce the problem formulation and provide a short overview of the state-of-the-art in debayering.

As shown in the last section, the data captured by a digital sensor can be described as a function of the incident radiance r_c, which depends on the spectral response L_c, which is defined by the IR and UV cut-off filters, the color filters for R, G and B and the sensitivity of the sensor's photo site (Eq. 2.1). The resulting image pixel values at the spatial sampling location n_1, n_2 additionally depend on the spatial response h, which is mainly determined by an optical low pass filter and the sensor's pixel pitch. We include an additive noise term N; however, the noise model will be discussed in detail in Sec. 2.2.

$$I_{Bayer}(n_1, n_2) = \iint\limits_{V} \int\limits_{\lambda} L_c(\lambda, n_1, n_2)\, r(u, v, \lambda)\, h(n_1 - u, n_2 - v)\, \mathrm{d}\lambda\, \mathrm{d}u\mathrm{d}v + N(n_1, n_2) \tag{2.2}$$

In a single-sensor camera, one filter type is placed at one spatial position, and the sensor output is a one-channel image. Thus, the filter function $L_c(\lambda)$ depends on the pixel's spatial position.

The task of debayering is to reconstruct the full-color data $I_{full-color}$ from the observed sensor data I_{Bayer}. An image example of the sensor data and the debayering result is shown in Fig. 2.2.

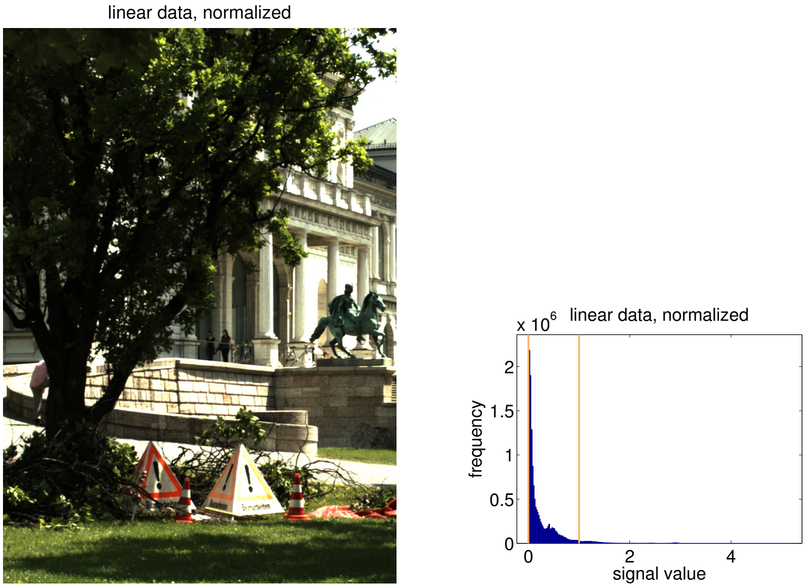
An overview on debayering techniques is provided by Gunturk et al. [48]. Available methods are classified according to the type (e.g., edge-directed or constant-hue-based interpolation) and the MSE is compared based on the low-resolution Kodak data set. A later survey and performance evaluation by Li et al. [71] classifies the methods into spatial-domain approaches and frequency-domain approaches. They investigate the performance of ten state-of-the-art approaches based on the low-resolution Kodak and the McM data set, which has a higher resolution. Li et al. claim the McM data set to be more challenging and closer to images acquired by digital cameras. Visually the images are clearly higher saturated; however, a proof for being more realistic is missing. Although the paper shows the high influence of the data set onto quality evaluation results, the need for further investigation on performance evaluation of debayering is identified. The final threat mentioned by Li et al. is that under these unrealistic testing conditions, "demosaicking alone might become a pet for academia only and never have any serious impact on the industry of electronic imaging" [71].

This lack of performance evaluation with true reference data is tackled by Andriani et al. [11] presenting a new dataset of real images including a method and a test image with a real full-color reference, which is obtained by a monochromatic sensor equipped with alternating color filters for every captured image. Thus, the above mentioned $I_{full-color}$ is available as a reference. However, the evaluation method is still imperfect because this camera system provides a solution for still image evaluation only. As the filters are altered subsequently over time, moving contents cannot be captured reliably and the setup is thus not sufficient for motion picture evaluation.

A set of ten different debayering algorithms is selected and used in this dissertation. The goal of selection is to have a representative set of algorithms, from very simple to very sophisticated, that represent the current state of the art in debayering. The simplest method is bilinear debayering, which is the basic interpolation using neighboring values of the same color. Additionally the following methods are used: the Matlab standard debayering[1], demosaicking with directional filtering and a posteriori decision (DDFAPD) [85], directional linear minimum mean square-error estimation (DLMMSE) [130], and the method of Lu and Tan presented in [77]. Additionally to the algorithms from literature, two camera debayering methods are used that were specifically designed for ARRI cameras, called Ada3 and Ada5. They can be utilized for processing Arriraw files, captured with an Arri camera, e.g., data from [11], with the freely available ARC software [2].

[1] Image Processing Toolbox, http://de.mathworks.com/help/images/ref/demosaic.html, 28.4.2015

[2] http://www.arri.de/BR/camera/alexa/tools/arriraw_converter/, 28.4.2015



(a)

(b)

Figure 2.4: Image example (a) and histogram (b) of a linear image. After normalization to the exposure index the main part of the signal values is fitted in the signal range [0,1], which is indicated by the yellow lines.

2.1.3 From linear data to monitor-ready image data

This section presents a general and simple procedure to obtain images displayable on a monitor from the linear data provided by the sensor [11]. The input data is the reconstructed RGB image in linear domain; thus, the pixel values have been interpolated using one of the available debayering methods, but they still are linear to the brightness in the scene.

A linear image directly displayed appears mostly black, only the highlights are visible. When the image is normalized according to the exposure, the main part of the image content can be displayed (Fig. 2.4a). The histograms of the images are displayed with the image examples. The normalized output data I_n is calculated from I_{raw} by adjusting the range from 16-bit-range (offset-subtracted) to $[0, 1]$ and correction depending on the exposure index EI with a multiplying factor. This normalization scales the values such that middle gray captured using an 18%-reflectance gray card, is mapped to the RGB values (0.18, 0.18, 0.18).

$$I_n = \frac{18EI}{400} \cdot \frac{I_{raw} - 256}{65535 - 256} \tag{2.3}$$

The next step is the reconstruction of the color values. To convert the color from camera RGB (R_c, G_c, B_c) to the usual sRGB monitor color space a color matrix conversion is applied. First, the data is converted into a wide gamut color space

(R_{wg}, G_{wg}, B_{wg}), whose primaries are chosen to avoid clipping in all but the most extreme cases.

$$\begin{bmatrix} R_{wg} \\ G_{wg} \\ B_{wg} \end{bmatrix} = \begin{bmatrix} 1.1766 & -0.119 & -0.0576 \\ -0.0194 & 1.0606 & -0.0412 \\ 0.0367 & -0.2019 & 1.1652 \end{bmatrix} \begin{bmatrix} R_c \\ G_c \\ B_c \end{bmatrix} \tag{2.4}$$

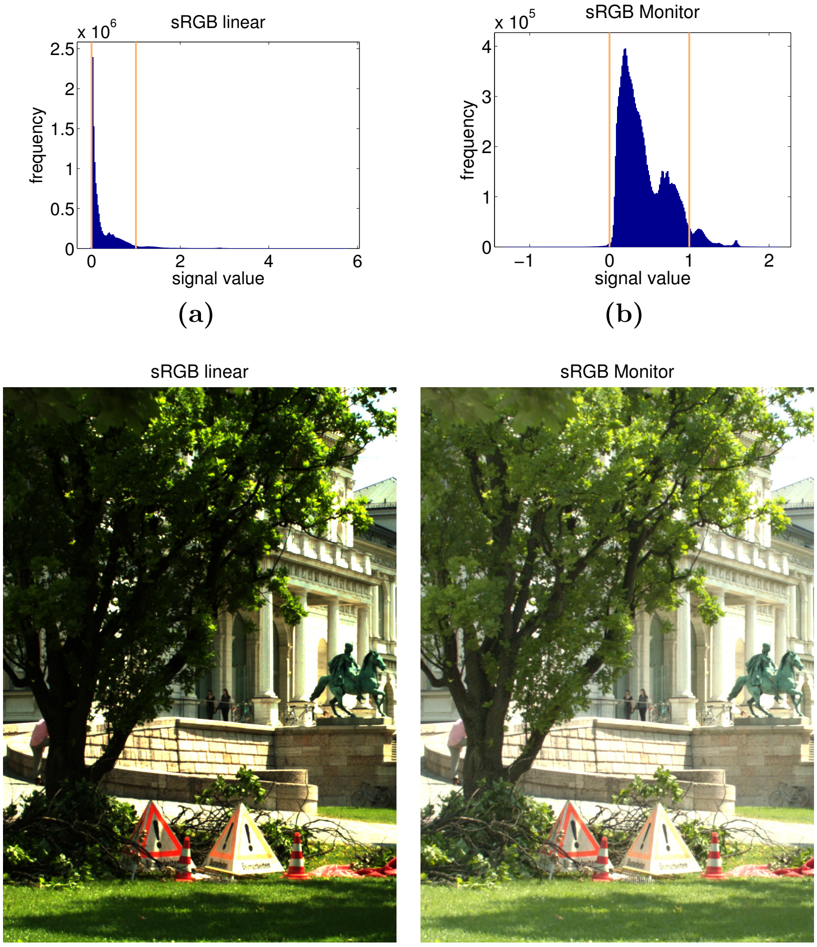
Subsequently, the conversion from the wide gamut color space to the color space defined in the ITU Recommendation 709 [56], which has the same primaries as sRGB, is performed using the following matrix.

$$\begin{bmatrix} R_s \\ G_s \\ B_s \end{bmatrix} = \begin{bmatrix} 1.6175 & -0.5373 & -0.0802 \\ -0.0706 & 1.3346 & -0.2640 \\ -0.0211 & -0.2270 & 1.2481 \end{bmatrix} \begin{bmatrix} R_{wg} \\ G_{wg} \\ B_{wg} \end{bmatrix} \tag{2.5}$$

The output sRGB images are displayed in Fig. 2.5c, the histogram in Fig. 2.5a. The last step missing for a monitor image is the compensation for the nonlinear electro-optical conversion function (EOCF) of the display device for $C = R$, G, or B, respectively), e.g., the transform described in the Amendment 1 of IEC 61966-2-1 [55].

$$C' = \begin{cases} 12.92\,C & \text{if } C \leq 0.0031308 \\ 1.055\,C^{1/2.4} - 0.055 & \text{if } C > 0.0031308 \end{cases} \tag{2.6}$$

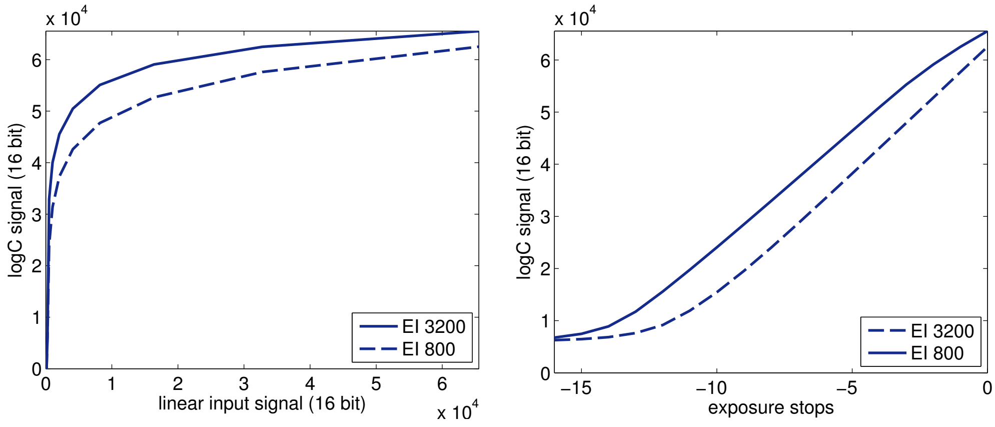
Fig. 2.5d shows the final display-domain image after the steps described here. Although this image has been converted correctly using the basic conversion steps, in the very bright areas the highlights are clipped. This may also be observed in the histogram of the output image in Fig. 2.5b. In the next section, a tone mapping operation is described, which allows a better highlight and shadow detail preservation.



(a) (b)

(c) (d)

Figure 2.5: After color space conversion, the linear sRGB data is obtained (a,c). Although the color values are corrected, the sRGB values additionally need a compensation for the monitor gamma. The result of this gamma compensation is shown in (b,d).



(a)

(b)

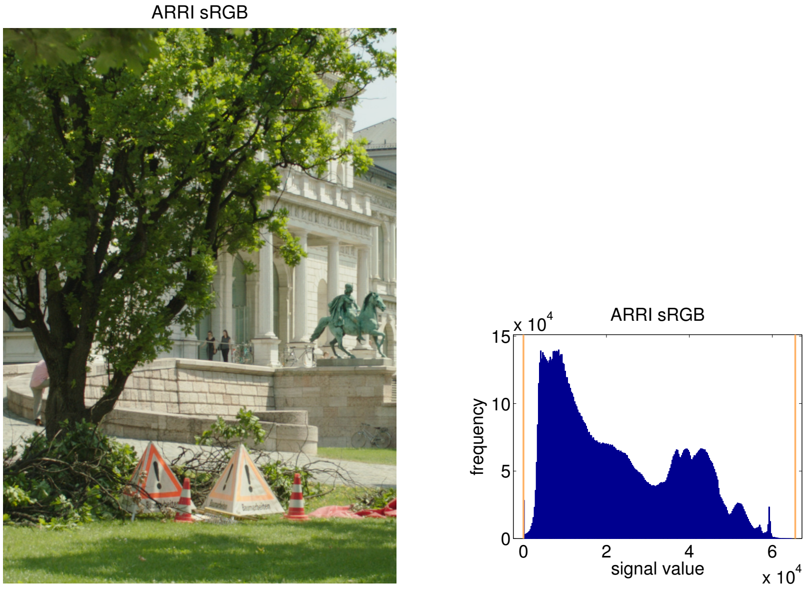
Figure 2.6: (a) shows the LogC curve for two different exposure indices plotted with linear x-axes. (b) shows the same curve with logarithmic x-axes corresponding to the exposure in stops (logarithm of x to the basis of two).

2.1.4 Nonlinear Transformations: Logarithmic Encoding and Tone Mapping

The basic processing described in the last section is not adequate for high-dynamic-range data delivered by today's image sensors. Fig. 2.7 shows the image when transformed using the camera processing procedure [134], which – additionally to the basic processing steps – includes non-linear transformations to better convert the full image information of the higher-dynamic-range linear image into the low-dynamic-range display domain representation.

The first nonlinear curve is a logarithmic transformation, called LogC transformation [135], which is applied in the wide gamut color space (Eq. 2.5). The LogC image is a format that can be captured with most ARRI cameras. Other camera manufacturers have similar logarithmic formats that allow preserving most image information but already provide visible image content when displayed on a monitor. The format was designed to match the characteristics of negative film: it is linear to the exposure over a wide range of signal values. The LogC curve, shown in Fig. 2.6, hence, transforms the linear data into a representation that is linear to the exposure around the middle of the signal range. To avoid clipping and to keep the information at the borders of the signal range, the curve flattens depending on the chosen exposure index.

The second nonlinear curve is the tone mapping, applied on the LogC data. Tone mapping is long known to be an important factor in reaching high photographic image quality. It compresses the highlights and shadows and provides a steep slope



(a) (b)

Figure 2.7: Monitor data, image example and histogram. With a nonlinear tone mapping function more pleasant shadows and more details in the highlights can be preserved.

in the main signal range, which leads to a high contrast monitor image [115]. The tone mapping curve, shown in Fig. 2.8, has a steep slope to increase the contrast matching the display specific range and compresses the highlights and shadows to avoid clipping.

An example of a LogC image is shown in Fig. 2.9 on the left, the resulting display domain image is shown on the right.

2.1.5 Filtering

Image sensors have almost always a size different from the output format, so an up- or down-scale filtering is required to rescale the images into the desired output format. The rescaling filter is an important choice, as the sharpness of the image can be adjusted in this step.

Image filtering can significantly influence the image characteristics. However, assuming that denoising is applied before rescaling, this part of the processing chain is not studied in detail.

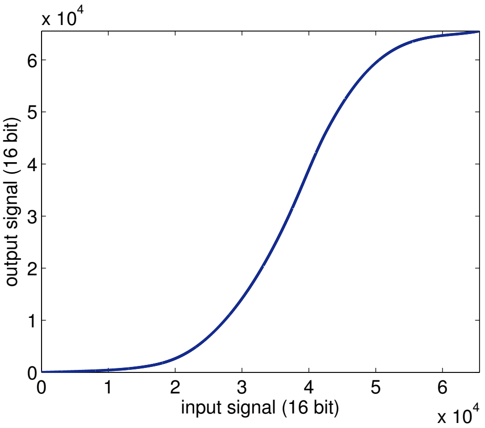


Figure 2.8: Tone Mapping curve

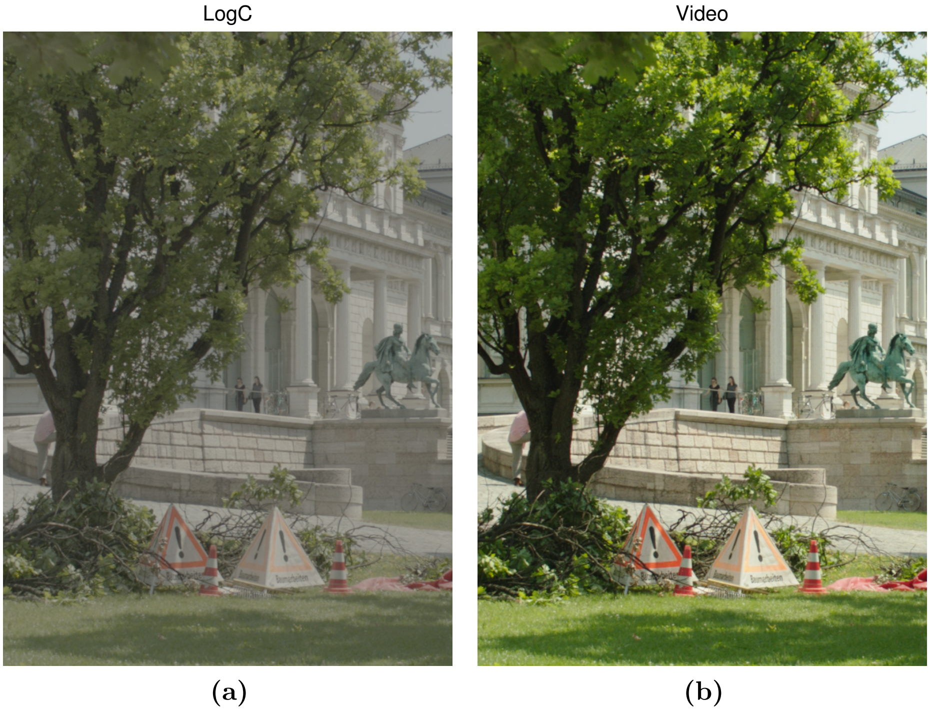


Figure 2.9: LogC image (a) and the final tone mapped monitor image (b).

2.2 Noise Characteristics of Real Camera Data

As explained in the last section, many steps are performed to achieve a color image ready for display, which can be summarized into the following three:

1. white balance,
2. debayering,
3. color transformations.

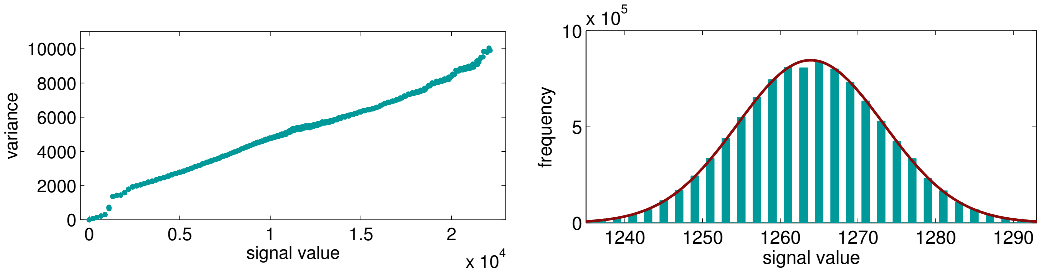
Applying these steps influences the noise in image data significantly.

To assess the noise characteristic changes caused by the processing steps, the noise variance is calculated. Two different approaches can be used for analyzing the noise variance: theoretically calculate the variance changes for all the operations applied in the processing steps or perform a Monte-Carlo simulation. We will use both approaches in the following sections and discuss the advantages and results.

Sec. 2.2.1 proposes a realistic model for the noise in the raw sensor data based on measurements. Noise in white-balanced images is discussed in Sec. 2.2.2. Debayering has a critical impact on the noise characteristics, and thus the noise after the debayering step is discussed regarding different aspects. Sec. 2.2.3 first discusses the visual quality, and subsequently, the signal properties, such as noise variances and spatial correlation, followed by an analysis of the noise in frequency domain and in wavelet domain. Sec. 2.2.4 evaluates the influence of the nonlinear color processing steps described in the previous section.

2.2.1 Noise in the sensor data

First, a realistic model for the camera noise in the raw sensor data is needed. Speaking of raw sensor data means calibrated Bayer sensor data, which is linear to the amount of light. To obtain reliable data for the realistic noise model, we measure the



(a) Variance **(b)** Distribution

Figure 2.10: Variance (a) and distribution (b) of the noise in the raw domain (signal values in 16 bit precision).

real camera noise in the raw sensor data based on a series of exposures. The noise variance is calculated using the photon transfer method proposed in the EMVA 1288 standard [138].

Although this measurement can be performed with any camera, in this dissertation, the ARRI ALEXA camera is used. As previously mentioned, the method can equivalently be applied with other cameras that capture raw data.

The photon transfer method [138] uses two subsequent frames recorded at constant and spatially homogenous lighting conditions. The temporal noise variance is calculated from the mean of the difference between two frames A and B in the active area of size $N \times M$.

$$\sigma^2 = \frac{1}{2NM} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (A_{ij} - B_{ij})^2 \tag{2.7}$$

This calculation of the variance is very robust to inhomogeneous lighting or fixed pattern noise because the variance is calculated directly from the pixel-wise difference of the frames without using the overall mean of the area $N \times M$. The corresponding mean signal value is calculated as the mean over all the signal values in the area $N \times M$ of these frames.

Fig. 2.10a shows the variance plotted over the respective mean signal value. The signal is the digital 16 bit value of the sensor output which represents the amount of light. The variance of the sensor noise can be approximated by a linear curve. This result matches the results with other cameras measured by Trussel et al. [121]. There is one difference though, in the signal region around value 0.1×10^4. The step in the variance curve is due to a special characteristic of the ALEXA sensor, the dual-gain read-out technology. The sensor read-out of the ALEXA provides two different paths with different amplifications (dual-gain read-out). The low-amplified path provides the data for the signal range starting from 0.1×10^4. The high-amplified path saturates in the high signal values, but for the low signal values it provides a significantly higher signal-to-noise ratio. The read-out noise (offset of the variance curve) is reduced, and hence, the dual-gain technology enhances the low-light performance of the camera. The two read-out paths are combined in the region around the signal value 0.1×10^4, which explains the step in the variance curve.

In Fig. 2.10b we show the distribution at a fixed signal level. The distribution is very similar to a Gaussian distribution, therefore, we can well approximate the sensor noise n in the raw domain using a Gaussian distribution with signal-dependent variance.

$$n \sim \mathcal{N}(0, \sigma(x)) \text{ with } \sigma^2(x) = m(x)x + t(x) \tag{2.8}$$

The variance $\sigma^2(x)$ is approximated as a piecewise linear function depending on the signal x, with the slope $m(x)$ and the intercept $t(x)$ based on the measurement data in Fig. 2.10a. Because of the dual-gain read-out the values for $m(x)$ and $t(x)$ are piecewise constant.

Other cameras do not show the step in the noise variance curve, but they show also a linear dependence of the noise variance on the signal value, due to the Poisson distribution of the photon shot noise. The fact that this is the dominant source of noise for modern image sensors has been shown for other cameras by Trussell et al. [121] and thus the presented noise model is also applicable in a similar way for other cameras.

Based on the proposed sensor noise model, the following sections evaluate the influence of the processing steps on the noise characteristics.

2.2.2 Noise after white balance

The first step, the white balance, is a linear gain factor g_c on each of the color channels. For a linear transformation, the output variance can be calculated.

$$\sigma^2(aX + b) = a^2\,\sigma^2(X) \tag{2.9}$$

For the mean value, the following relationship holds.

$$E[aX + b] = aE[X] + b \tag{2.10}$$

The white balance directly influences the noise level n_c in the different colors.

$$n_c = g_c\,n \sim \mathcal{N}(0, g_c\sigma(x)) \tag{2.11}$$

The white balance changes the noise level of the different channels according to the gain factors g_c. This leads to a different variance curve for the color channels, as it can be seen in Fig. 2.11. As described in Sec. 2.1.1, these factors depend on the lighting conditions in the scene, and are chosen based on the color temperature.

Fig. 2.12 shows the effect over the range of color temperatures for a assumed input noise level of one. The most common lighting conditions are daylight, corresponding to 5600 K, and artificial light, usually corresponding to 3200 K.

As most denoising algorithms adjust the parameters depending on the noise level, denoising algorithms can be adapted to the different level by choosing a different denoising strength for each channel.

2.2.3 Noise after debayering

The debayering is the second step of the processing pipeline. It generates three color values by interpolation using the pixel value and the neighboring pixel's values. Debayering introduces spatial and chromatic correlation in the noise, as it exploits the correlations present in the signal for estimating the missing pixel values. The

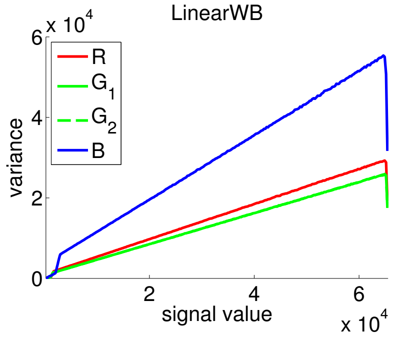


Figure 2.11: Noise varaince over mean signal value after the white balance (3200K).

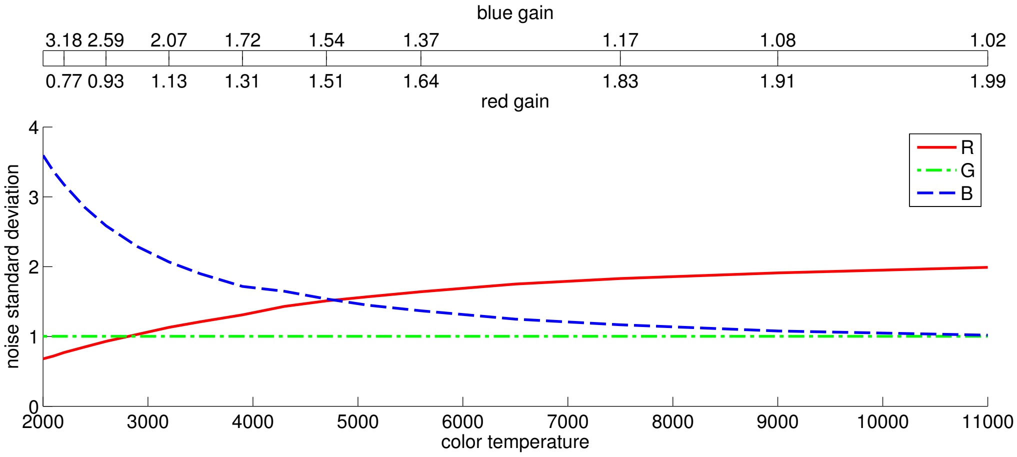


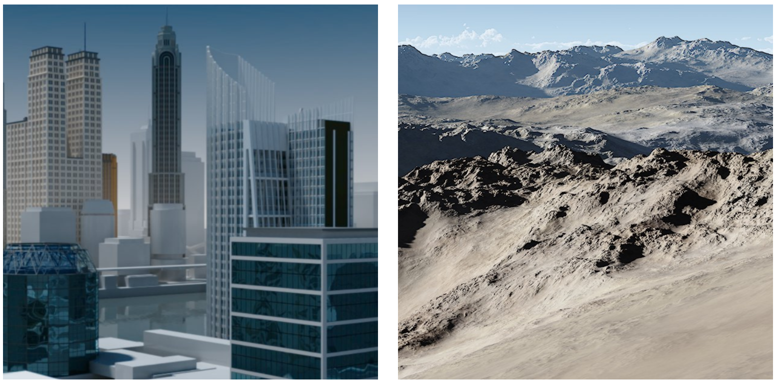
Figure 2.12: The noise standard deviation after white balance is plotted over the color temperature. The input noise was zero-mean Gaussian noise with $\sigma = 1$.

spatial correlation of the noise n_{deb} in the debayered image is usually disregarded in common state-of-the-art denoising methods [5].

Debayering significantly changes the noise characteristics. The visual appearance is discussed firstly to evaluate the impact of debayering on the visual image quality. Subsequently, the influence of debayering on noise characteristics is analyzed in three different domains: the spatial domain (or pixel domain), in the frequency domain and finally in the wavelet domain. The content is partially based on previous work [1, 5, 10]. Additionally, we analyze the frequency spectrum of film grain noise and compare it to the digital camera noise after debayering.

Visual Perception of spatially correlated noise

To evaluate the noise characteristics after debayering, our test setup includes a reference image and a noisy image with different types of noise. Although noisy images could be obtained in the form of real camera data, we would lack the corresponding



(a) city **(b)** landscape

Figure 2.13: The computer-generated test sequences

reference imagery. Adding noise to a readily processed image, e.g., an image from the Kodak data set, can be used for first evaluations, but is not optimal for tests with realistic camera noise, as it is necessary to include all the processing steps described in the previous section. A full-color reference in the linear domain is therefore required to obtain realistic test data. We obtained this reference data from a rendered 3D scene.

Two scenes are used, see Fig. 2.13. The frames are rendered in high resolution and in linear signal domain. To incorporate the optics of a camera system, the images are multiplied in the Fourier domain with the optical transfer function of the camera. This step takes into account the diffraction-limited lens, the optical low-pass filter and the pixel aperture, as described by Schöberl et al. [109]. The camera noise is added in linear domain according to the measured values presented in Sec. 2.2.1 using a Gaussian distribution with a signal-dependent variance defined by the linear approximation of the measurement data. This provides simulated raw images with a reference. For more details regarding the reference data refer to Sec. 5.1.1 in the chapter discussing denoising evaluation.

To assess the effect of debayering on the noise characteristic, the image processing with debayering is compared to the processing without debayering, which is only possible with the simulated RGB values; in real single-senor camera images, the debayering cannot be omitted. A crop of the city sequence with noise and with-/without debayering is shown in Fig. 2.14. The images (noisy, with and without debayering) are compared to the reference. The scaled differences of the respective noisy image to reference image, shown in Fig. 2.14, visualize the effect: when debayering is included the noise is structured, with a coarser grain, and more colorful. As the contrast sensitivity of the human visual system (HVS) shows a low-pass characteristic in the color vision channels, a shift to lower frequencies can increase the visibility of the color noise.

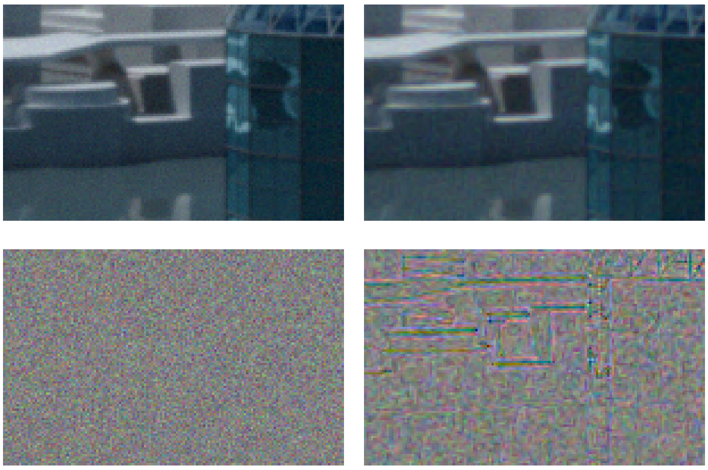


Figure 2.14: Crop of the sequence “City”. Noisy image (left) and noisy image with debayering (right). In the second row the respective difference image I_d scaled for display ($I_{d,\text{scaled}} = I_d \cdot 4 + 128$).

To obtain reliable information about the human perception of the different noise characteristics, the data of a subjective test provides more information. 18 participants rated the quality of noisy video sequences on a discrete scale from one to ten. The test was performed in the ITU-R BT.500 compliant video quality evaluation laboratory, for details about the test setup refer to Chapter 5. From the votes, the mean opinion score (MOS) is calculated, which then can be used as measure of the perceived visual quality.

Four different noise models are compared: the usual AWGN model, AWGN with debayering, signal-dependent noise without debayering and finally the realistic camera noise model – signal-dependent noise with debayering. Based on the MOS, we evaluate the visual quality of the noisy test sequences. The spatially correlated noise is perceived as more annoying. Although the MOS is different depending on image content and noise type, Fig. 2.15 shows a lower MOS for the all the demosaicked sequences compared to the sequences with uncorrelated noise. The spatially correlated noise, which shows coarser grain and appears more colorful, hence, is more disturbing. The MOS of the city sequence with AWGN is about 3 scores lower when debayering is included. Regarding the sequences with signal-dependent noise, the MOS is 0.5 lower for the city sequence and 2.3 lower for the landscape sequence when debayering is included. We thus showed the significant effect of the noise characteristics on the visual perception of color video sequences: the spatial correlation of the noise decreases the perceived image quality, and hence, the debayering has a significant effect on the visual perception of noise in video sequences. In the next section, the technical details of the noise characteristics are discussed.

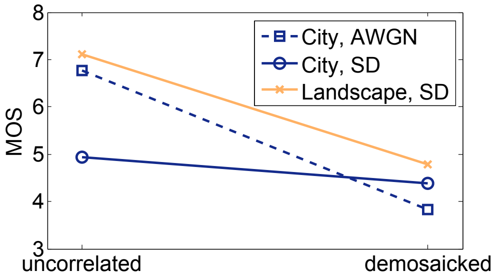


Figure 2.15: The MOS results for the test sequences "City" and "Landscape" using the traditional AWGN model (dashed) and the realistic signal-dependent noise (solid lines). The uncorrelated noise, processed without debayering, is shown on the left, the results with debayering on the right.

Analysis of the noise after debayering: local correlation and noise reduction

Two different approaches can be used for analyzing the noise variance: theoretically calculate the variance changes through the operations applied in the processing steps or perform a Monte-Carlo simulation. We will explain shortly how the calculation can be performed for bilinear debayering.

The goal of the theoretical calculation of the variance is to find the variance of the output noise n_{deb} based on the variance of the input noise n_c. For a linear transformation, the output variance can be calculated as given by Eq. 2.9. To investigate the influence of debayering mathematically, the most simple debayering algorithm is studied: the bilinear interpolation. Bilinear debayering can be considered as a linear transformation. Although the debayering algorithms used in state-of-the-art image processing are usually more sophisticated, the principal method and can be shown using this simple debayering method.

Noise variance after bilinear debayering The bilinear interpolation of a green pixel value $g_{i,j}$ is the mean of the neighboring green values.

$$g_{i,j} = \frac{1}{4}\left(g_{i-1,j} + g_{i+1,j} + g_{i,j-1} + g_{i,j+1}\right) \tag{2.12}$$

Assuming the variance of all green samples is the same, the variance of the interpolated green value is $\sigma^2(g_{i,j}) = \frac{1}{4}\sigma^2(g_{sampled})$. This leads to a lower variance after debayering. A similar investigation can be done for the red and blue interpolation. As we have 50% of green sample values and 50% of green interpolated values the overall green variance should be $0.5(\frac{1}{4}\sigma_g^2 + \sigma_g^2) = 0.625\sigma_g^2$. This matches the value obtained by simulation for noise in one image (720 × 1280 pixels), shown in Tab. 2.1. In this table the numbers are accurate to two decimal places.

The theoretical calculation can easily be performed for the bilinear debayering method, which is a very simple debayering method. The theoretical calculation provides precise results, and the results of the simulation generates sufficiently precise

Table 2.1: Noise variance after Debayering step. With σ_n^2 the noise variance before Debayering step is indicated.

R channel	G channel	B channel
0.56 σ_n^2	0.62 σ_n^2	0.56 σ_n^2

results, too. The remaining question is, whether both methods can be used for other debayering methods. The variance calculation after one linear transformation is simple, but most debayering algorithms consist of many steps, which make the calculations more complicated. Additionally, these steps can be nonlinear, thus linear approximations are necessary.

Noise variance after camera debayering The camera debayering (Ada3) is an example of a more complicated debayering method, as it performs many steps. For the camera debayering, the output variance has been estimated step by step, based on the assumption of homogenous regions. This is a valid assumption for the improvement of denoising algorithms since the noise is more visible in homogeneous image regions than in textured regions. As the nonlinear steps need to be approximated, and the correlation is not taken into account, small errors accumulate in the calculations. The overall error is signal-dependent and color dependent, and it can be significant for some signal values, especially for saturated color values, whereas the simulation provides precise results for all signal values due to the large image size used in the simulation (HD). Additionally, the simulation provides the possibility to generate results for algorithms that are not published. This is a clear advantage when camera data is considered as debayering algorithms for camera data are usually not published. Based on these considerations, the simulation is the better choice, as it is precise enough and more flexible.

The analysis above shows that the variance of the noise is lower after debayering. This result – debayering reduces the noise – may seem surprising, as the debayering does not aim a noise reduction and the visual quality was shown to decrease with debayering in the last section. Although it is correct that debayering technically reduces the noise, this does not lead to a more visually appealing image, as debayering errors are introduced and the noise characteristic is changed. To understand the reasons, we need to study not only the variance of single pixels, but also the correlation between the pixels.

Spatial correlation of the noise after debayering To evaluate the influence of debayering for different methods, we first use the standard test images from Kodak. This choice is due to the fact that most debayering algorithms in literature are optimized using that test set, but the results also hold for other image data.

Table 2.2: Correlation matrices for different debayering methods, calculated using AWGN and the Kodak data set.

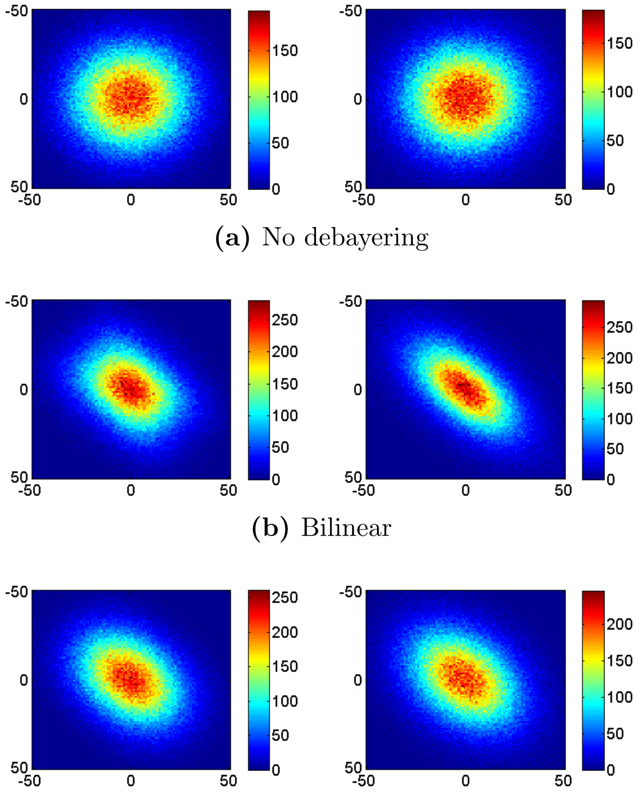
$C_{bilin} =$	$C_{DLMMSE} =$
$\begin{bmatrix} 1 & 0.5693 & 0.1243 \\ 0.5715 & 0.3214 & 0.07181 \\ 0.1284 & 0.07425 & 0.0191 \end{bmatrix}$	$\begin{bmatrix} 1 & 0.2239 & -0.0066 \\ 0.375 & 0.06351 & 0.03626 \\ -0.003771 & 0.03715 & 0.03545 \end{bmatrix}$

The spatial correlation of the noise is evaluated after debayering a noisy and a noise-free image. The difference image contains the error introduced by the noise. We use this difference image to calculate 2D-histograms of the noise distribution and the correlation matrix C for different debayering algorithms.

To visualize the distribution we plotted 2D-histograms of the noise. The difference of the noisy and the reference image is the error due to noise. In the 2D-histogram the error value occurrence for two neighboring pixels are plotted. The color represents the frequency (number of occurrences); the position in the plot represents the value of the error. The resulting scatter plots for an image with AWGN without debayering are given in Fig. 2.16a. The distribution is symmetric as expected for uncorrelated noise. Fig. 2.16b and 2.16c show the scatter plots of the noise after debayering using bilinear interpolation and using the debayering method proposed by Lu/Tan [77]. The scatter plots of the debayered noise are dispersed into the diagonal direction, which indicates a correlation: it is more likely for a pixel to have the same or a similar error value as its neighbors. The R channel is not shown; since the debayering methods interpolate the B and R channels the same way, the scatter plots of B and R look very similar.

We calculated the correlation matrices C, given in Tab. 2.2. The matrices contain the correlation between a pixel (i, j) and its neighbors; more precisely the entry (k, l) in the matrix corresponds to the correlation of the neighbor pixel $(i+k, j+l)$. Numbers for bilinear interpolation and DLMMSE [130] are given. Whereas in the usual uncorrelated noise model the noise values don't correlate with the neighboring values and thus lead to a matrix with only a one in the upper left matrix position, these correlation matrices show that a strong correlation to the neighboring pixels is present after debayering. The matrices are 3×3 because numbers outside of this regions are very small. The bilinear interpolation has the strongest correlation.

The correlation implies that spatial denoising algorithms that rely on neighboring values will not work properly on debayered data. Denoising methods operating on Bayer data are thus of great interest.



(a) No debayering

(b) Bilinear

(c) Lu/Tan debayering

Figure 2.16: Scatter Plot of the noise in a noisy Kodak image before (a) and after debayering (b,c); the first Kodak image was used and AWGN with $\sigma = 20$ added. From left to right the G and B channel.

Noise in Fourier domain and comparison with film grain

Whereas the last section described the influence of debayering on the noise characteristics in the pixel domain, the following analysis provides details about the noise characteristics in the frequency domain. The frequency spectrum is important for the human vision; due to the contrast sensitivity function (CSF), very high frequencies are less visible than middle or low frequencies. As film grain noise is considered to be more pleasant than today's digital camera noise, we compare the digital camera noise spectrum to the spectrum of scanned film grain.

We compare the noise power spectrum, which is the squared amplitude of the transformed noise. For the camera noise, we use the noise model of the sensor noise as it was described in Sec. 2.2.1 and apply the processing chain described in Sec. 2.1.1 – Sec. 2.1.4 to obtain noisy LogC images. The film grain is obtained by capturing a grey card, thus a homogenous area, with different film types. The film is scanned to obtain a digital representation with the same resolution as its digital counterpart. We subsequently calculate the noise power spectrum for both the scanned film grain and the simulated camera noise using the same procedure. First, we remove the overall mean of the noisy image I_{sp} with the image size $N \times M$.

$$N_{sp} = I_{sp} - \frac{1}{NM} \sum_{i,j=0}^{N,M} I_{sp}(i,j) \tag{2.13}$$

The noisy image is transformed using an FFT algorithm to obtain the noise spectrum $F(\omega)$.

$$F(\omega) = \mathcal{F}(I_{sp}) \tag{2.14}$$

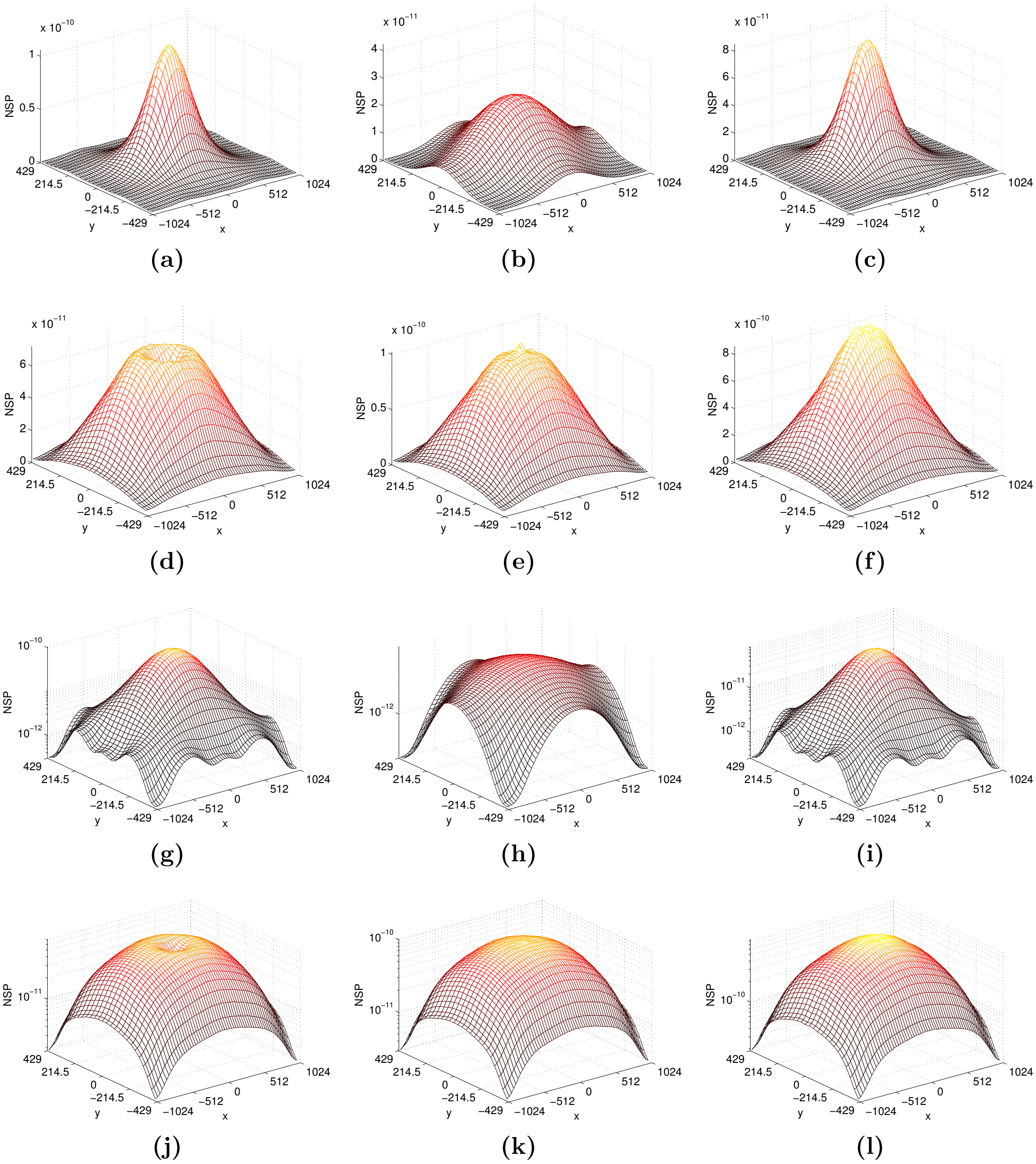
The noise power spectrum is the squared amplitude normalized by the image size.

$$NPS(k,l) = \frac{1}{NM} |F(k,l)|^2 \tag{2.15}$$

For the mesh plots of the noise power spectrum, the median per 32 × 32 block is calculated and plotted in Fig. 2.17.

The spectrum of the digital camera noise is clearly not white; it shows a peak towards low frequencies. This probably makes camera noise more visible for the HVS than white noise. Camera noise additionally shows a higher power spectrum for vertical and horizontal high frequencies than for diagonal high frequencies, especially in the green channel. Both results show that the noise spectrum is not white, but the film grain spectrum is broader than the spectrum of the digital camera noise and less direction dependent. This might explain that film grain noise is more accepted and sometimes even appreciated by artists. The noise power spectrum of film grain noise depends on the type of film material used. Five film types were analyzed and the most representative is shown in Fig. 2.17.

Although the analysis of the noise power spectrum is interesting for explaining the human perception, the frequency domain is not optimal for efficient camera



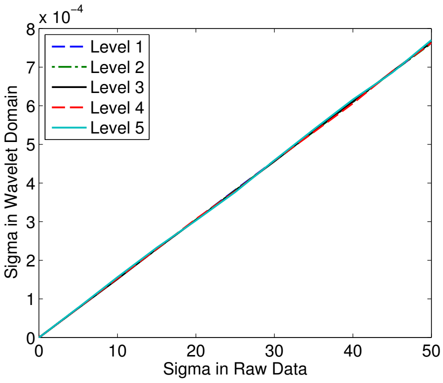
(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 2.17: Noise power spectra of digital camera noise and film grain noise: (a), (b), and (c) show the digital camera noise spectrum in the R, G and B channel respectively. (d-f) show the noise power spectrum for film grain in R, G, and B channel respectively (Kodak Vision3 500T color negative film 5219). (g)-(l) show the data from (a)-(f) with a base-10 logarithmic scale used for the z-axis.



(a) no deb.

Figure 2.18: The noise variance of the wavelet coefficients over the input variance is shown for AWGN.

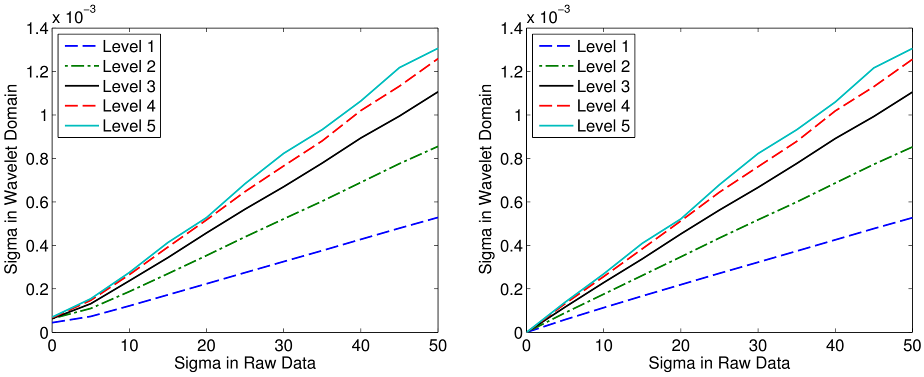
denoising. Kokaram showed in [64] that a 3D-FFT based method can be used for denoising non-white noise, but the method is not a good choice when the target is a stream-based video processing pipeline since the complete image is stored in the memory before the FFT can be applied. We therefore choose a second representation for our analysis, the wavelet domain, to obtain a description of the camera noise that might be more useful for developing denoising methods. Wavelet domain coefficients do provide frequency information, too, but additionally they are localized and can be computed based on a limited spatial neighborhood.

Noise in wavelet domain

This section analyzes the noise characteristics of wavelet coefficients when camera noise is present in the images. This investigation is important for three reasons. First, wavelet methods seem specifically practical for denoising camera data: whereas it may be difficult to include the spatial correlation discussed in the last section in pixel-domain denoising methods, wavelet-domain denoising methods are a good choice, because the spatial correlation leads to a different amount of noise in the different frequency bands, which are represented by wavelet levels. Second, wavelet denoising methods have generally shown good results, and third, they can be expected to be implemented quite efficiently for video processing. Parts of the following analysis has already been presented in [1].

The commonly used noise model AWGN has a white noise distribution in frequency domain, i.e., it is equally strong in all frequency bands. Conversly, camera noise is not white due to the spatial correlation. In wavelet domain, this means that the noise is level-dependent.

With a Haar filter, the wavelet filter type with the lowest complexity, the filtering operation corresponds to a mean and a difference. For an uncorrelated input, the



(a) RGB reference

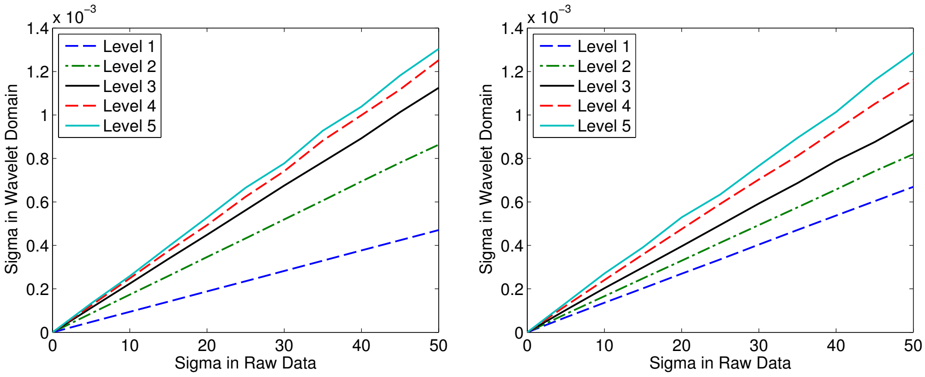
(b) debayered reference

Figure 2.19: In both plots, the noise standard deviation in the wavelet domain is plotted over the input noise standard deviation. In (a) the full RGB reference was used. This means, the plot shows the noise plus an additional debayering error. In (b) the reference image was debayered, therefore the debayering error is not included and all lines start at zero. Camera debayering method Ada3 was used.

variance of the filter outputs can be easily calculated. However, the signal is spatially correlated due to debayering before the transformation.

To show the effect of debayering on the noise variance in the wavelet levels, we therefore perform a simulation. Artificial AWGN is applied to the computer-generated City image (described in Sec. 2.2.3) and the standard deviation of the wavelet coefficients is plotted for the different wavelet bands in Fig. 2.18. As expected for the commonly used white noise model, the noise is equally strong in all frequency bands. Subsequently, we used the artificial image again, but this time applied a Bayer pattern to obtain sensor comparable data and applied debayering to obtain the debayered noisy image. Fig. 2.19 shows that the noise in the high frequency bands (Level 1 and 2) is diminished by debayering, because of the interpolation step. The noise in the higher levels is slightly stronger. This matches well the frequency distribution shown in the last section. Fig. 2.19 shows two slightly different plots. The difference is the reference image, which was used to calculate the noise standard deviation in the wavelet domain. For the plot in Fig. 2.19a, the full RGB reference was used, which is available for the simulated city image. This means, the plot shows the noise plus an additional debayering error. In Fig. 2.19b, the reference image has been debayered before comparison, thus the debayering error is not included. As in this dissertation denoising is the target, the second method is used in the following plots.

To find out whether the described effect is specifically valid for one debayering method or for more debayering methods in general, a set of ten different debayering



(a) Bilinear **(b)** DLMMSE

Figure 2.20: In both plots, the noise standard deviation in the wavelet domain is plotted over the input noise standard deviation. In (a) bilinear debayering was used. In (b) DLMMSE [130] debayering was used.

algorithms is used: bilinear debayering, camera debayering (Ada3 and Ada5 [3]), DDFAPD (Demosaicing With Directional Filtering and a posteriori Decision) [85]), DLMMSE (directional linear minimum mean square-error estimation) [130], LPA-ICI [96], Matlab standard debayering [4], POCS (projections onto convex sets [47]), RI (residual interpolation [89]). The results show that all tested debayering methods lead to very similar wavelet noise characteristics. The order of the noise strength in the wavelet levels is the same; level 1 and 2 have clearly the lowest noise standard deviation, and the noise level rises with the number of level. DLMMSE shows the least spread between the levels. The result for bilinear debayering and DLMMSE is shown in Fig. 2.20.

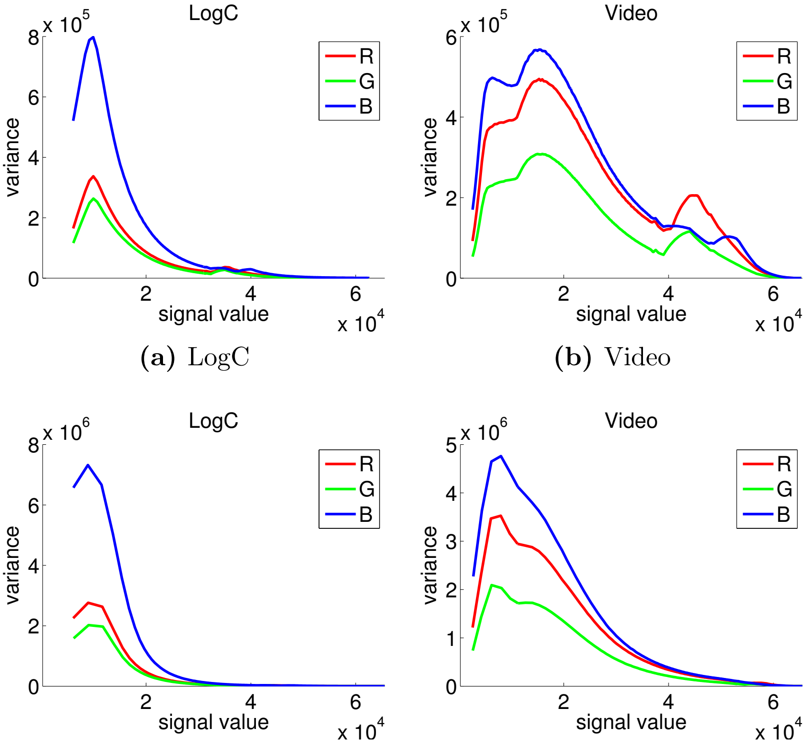
These results can be very helpful for designing denoising algorithms that operate in wavelet domain.

2.2.4 Noise after color transformations

The third step, the color transformations, includes a nonlinear tone mapping and color space conversion to map the linear values to displayable signals. Color transformations can strengthen the spatial and chromatic correlation of the noise. This section discusses the processing as presented in Sec. 2.1.3 and Sec. 2.1.4, but the exact way color conversion is done is an individual choice, and the presented results of noise variance and distribution only hold for the processing described. However, for today's high-dynamic-range image sensors tone reproduction operations similar

[3] http://www.arri.de/BR/camera/alexa/tools/arriraw_converter/, 28.4.2015

[4] Image Processing Toolbox, http://de.mathworks.com/help/images/ref/demosaic.html, 28.4.2015



(a) LogC

(b) Video

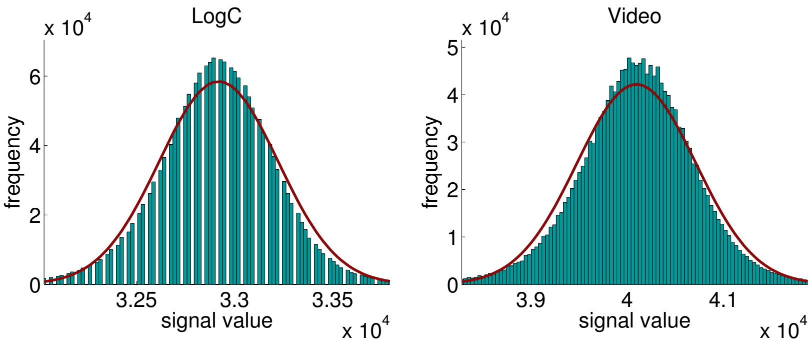
(c) LogC

(d) Video

Figure 2.21: The noise variance over mean signal value is plotted for LogC data and video data and for EI 800 (top) and EI 3200 (bottom). In the EI 3200 the maximal variance is almost ten times higher than for EI 800.

to the one described are common, as they are necessary to obtain a visually pleasant image on a low dynamic range display device. The transformations are not new, tone mapping operations have already been studied for analog photography, e.g., in [115].

For the processing described in Sec. 2.1.4 the noise variance is be calculated in the LogC and the video data for different mean signal levels. Based on the model for sensor data described in Sec. 2.2.1, noisy image sensor data is generated. Subsequently this data is transformed to LogC and then to display domain data. To obtain LogC data, the wide gamut matrix transformation is used followed by the nonlinear logarithmic transformation, the LogC curve. For the display domain output, the tone mapping, conversion to sRGB color space and the gamma correction is additionally applied. This simulation is performed for stepwise increasing signal levels in the linear data to obtain a sufficiently detailed variance curve. 480 signal values have been used for the LogC and video domain. We use two different parameters for the exposure index, 800 and 3200. Exposure index (EI) 800 usually means good image quality with acceptable noise level; EI 3200 is used for low light



(a) LogC **(b)** Video

Figure 2.22: Noise distribution after color transformation.

conditions and contains a significant amount of noise.

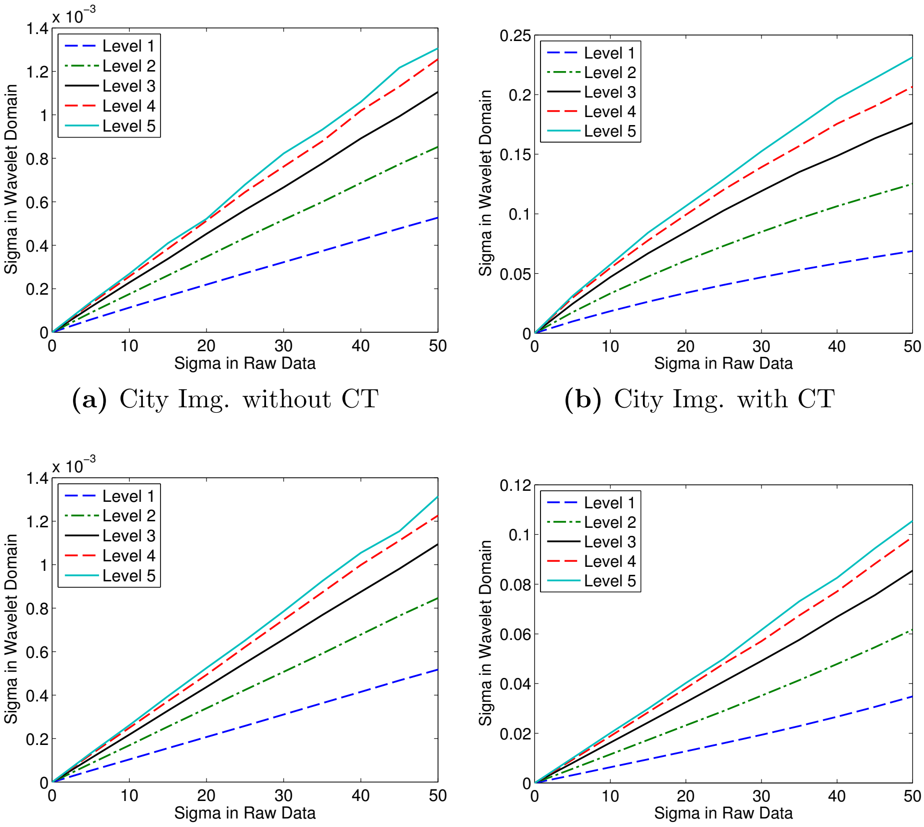
Fig. 2.21 shows the results. The most important finding is that the variance in the display domain and LogC data does not increase with signal level, as it was the case for linear sensor data (Sec.2.2.1). It even shows the opposite tendency: after a peak at low signal levels, the variance decreases with increasing signal levels. Since most of today's algorithms for image processing are designed and tested based on readily processed image data, which means image data corresponding to the display domain image, this finding is very important. It shows that it is very unrealistic to test and design methods for real sensor noise (e.g. the Poisson noise reduction in [116]) based on readily processed data. We expect that our publication of a real camera image data set [11] can serve as more realistic test data for such methods in future research.

The last step is to evaluate the noise distribution. Whereas the distribution in the raw data matches a Gaussian distribution (shown in Fig. 2.10b), the LogC and video data show a distribution that is more compact around the mean value and has longer tails. The representative distribution for one signal level is shown in Fig. 2.22a and Fig. 2.22b.

In summary, the results show that the noise after the color transformations is significantly different from the noise in the linear data: the distribution does not fit a Gaussian distribution and the variance is not linearly dependent on the signal. In contrast to the sensor noise, the variance is lowest for high signal levels and not for low signal values.

Noise in wavelet domain

To evaluate how the different noise characteristics after color transformations influence noisy wavelet coefficients, the noise in the wavelet coefficients on two images with and without color transformations is analyzed. Fig. 2.23a shows the results for the city image without color transformations; the results including color transfor-



(a) City Img. without CT

(b) City Img. with CT

(c) Grey Img. without CT

(d) Grey Img. with CT

Figure 2.23: The noise standard deviation in the wavelet domain is plotted over the input noise standard deviation.

mations are shown in Fig. 2.23b. Comparing Fig. 2.23a with Fig. 2.23b shows that including the color transformations leads to a nonlinear relationship between input noise (σ in the raw data) and wavelet domain noise (σ of the wavelet coefficients of the transformed display domain image).

Two images are used, a homogenous grey image and the city image. On the grey image, i.e., homogenous image content with a fixed signal level, the curve is almost linear (Fig. 2.23d) since the color transformations do not change abruptly and can be well approximated by a linear transformation around a fixed signal level. However, the results show a higher slope towards higher sigma levels. This nonlinearity can be explained by very noisy values being far away from the signal value, where the linear approximation of the color transformations does not hold. Here the gain is higher than at the mean signal level and therefore the values have higher noise than expected based on a linear approximation at the mean signal level. For visually pleasant denoising results, it is sufficient to use an approximation that holds in homogenous areas, as noise is less visible in textured image regions. However, this is only the case if the noise in structured regions or at edges tends to be higher than expected based on the approximation, because this leads to lower denoising strength than the true variance would require, which is usually fine in structured areas and does not cause artifacts. However, as the noise level depends on the signal, this may not be fulfilled and could lead to denoising artifacts that decrease the image quality. Denoising algorithms therefore are best applied in either the sensor data or the linear debayered data.

2.3 Denoising in the camera processing pipeline

Based on the last section's analysis, we know that the noise of camera data is not Gaussian, and with every processing step it veers away more from the usually assumed AWGN model. To answer the question what impact the difference between the traditional independent noise model and real camera noise has, the effect on denoising is evaluated in this section. Two main questions are discussed: how do the noise characteristics influence the performance of state-of-the-art algorithms? Which domain, e.g., linear, LogC, or display domain, is best suited for denoising?

Whereas firt results, drawing from a previously published paper [5], are presented to discuss these questions in more detail already in this introduction, the remainder of this section provides an overview on existing denoising methods. Results on camera data are presented and adaptions of the algorithms for camera data are proposed and evaluated.

Denoising images containing more realistic noise

The first question is how the noise characteristic influences the performance of today's denoising algorithms. Do they show very different results? To answer this question, we apply two state-of-the-art methods on the city image. First, in the "traditional way", i.e., AWGN is added directly in display domain. Second, in a more realistic camera-like way, i.e., signal-dependent noise based on the model presented in Sec. 2.2.1 is added to the Bayer data and then the image is processed till display domain. Additionally, different versions were generated to find out which properties of camera noise have the biggest impact on denoising effectiveness, e.g., signal dependence or debayering.

The PSNR of the denoising results is shown in Tab. 2.3. We compare signal-dependent camera noise added in linear domain based on the model described in Sec. 2.2.1, to Gaussian noise added in display domain. The variance of the Gaussian

Table 2.3: PSNR values for denoising results of the city sequence with BLS-GSM [101] and BM3D [33] are shown. Two noise types are used: Gaussian noise added in display domain (AWGN) and signal-dependent camera noise added in linear domain (SD). Denoising is either performed in linear domain or in display domain.

	Noisy		BLS-GSM		BM3D	
	RGB	deb.	RGB	deb.	RGB	deb.
AWGN / display-domain denoising	35.51	35.35	44.47	38.81	46.33	38.17
SD / display-domain denoising	31.51	33.95	36.45	37.43	43.11	39.49
SD / linear-domain denoising	31.51	33.95	39.40	36.61	41.99	38.34

noise was set to obtain images giving a similar visual impression in display domain. We process the frames in two ways: without debayering (called “RGB” in Tab. 2.3) and with debayering (“deb.” in Tab. 2.3). Note that the RGB version is only possible as we have the computer-generated full-reference RGB values; in real raw images the debayering cannot be omitted. Subsequently, all these versions are denoised, either in the linear domain or in the display domain. Two state-of-the-art denoising algorithms are used: BLS-GSM [101] and BM3D [33]. The parameters are picked to obtain the most visually appealing results. We calculated the PSNR of each frame and took the mean over 20 frames.

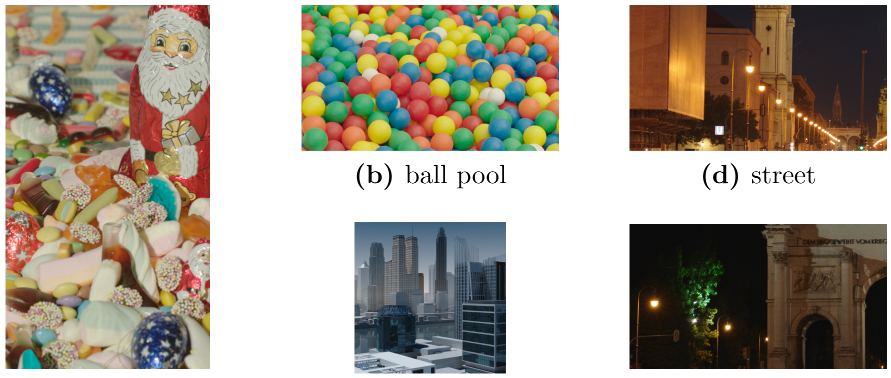
The effect of debayering on denoising effectiveness is evaluated. Although the PSNR of the noisy image sequence after debayering is a bit higher than without debayering – except a similar PSNR for the unrealistic case of AWGN debayered in display domain – denoising results of debayered noisy data do not show higher PSNR. BM3D shows significantly lower PSNRs when applied after debayering. BLS-GSM also shows a significantly lower PSNR on debayered data compared to AWGN. On signal dependent noise the PSNR is not lower, but the PSNR gain of 5 dB for uncorrelated noise drops to 3.5 dB for debayered noise. These results thus show that spatially correlated noise due to debayering is more difficult to denoise.

We now compare denoising effectiveness on AWGN and on signal-dependent noise. BM3D seems to be quite robust to the signal-dependence; it leads to an improvement of above 10 dB in PSNR for both constant AWGN and uncorrelated signal-dependent noise. BLS-GSM, which showed an improvement of 9 dB for AWGN, provides lower PSNR improvement for the signal-dependent noise (5 dB), and thus is less robust than BM3D. We can conclude that the effect depends on the algorithm.

Finally, we compare the realistic camera noise, including both signal-dependence and debayering, to the usual AWGN model. The PSNR of BM3D for the AWGN version is 46.33, which corresponds to an improvement of more than 10 dB (for BLS-GSM 9 db). The realistic camera noise can only be reduced by around 4 dB with both algorithms. These results, based on the city image, indicate that the question of whether existing denoising algorithms show very different results can be answered with yes; denoising algorithms show very different results on data with camera noise characteristics.

Denoising domain

What is the best domain for denoising with a state-of-the-art algorithm, linear, logarithmic or the display domain? While denoising can be expected to work well in a data domain that is close to the human perception, it can also be expected to work well in a domain with known noise characteristics. This, however, cannot be achieved together, and thus a trade-off between the two requirements must be found. To answer the question which domain is best suited for denoising, BM3D and BLS-GSM denoising were applied in linear domain.



(a) candy

(b) ball pool

(c) city

(d) street

(e) arch

Figure 2.24: Test sequences for denoising evaluation

BM3D achieves higher PSNR in display domain both on RGB data containing uncorrelated noise and on debayered data. BLS-GSM achieves higher PSNR in linear domain on RGB, but the PSNR is higher for display domain denoising on debayered data. Hence, no clear answer can be found based on that data.

The above presented results based on the city image sequence and two state-of-the-art algorithms indicate that results can depend on the denoising method and on the denoising domain. Therefore in the following, more different denoising methods will be discussed.

Denoising methods

Denoising methods can be classified into pixel domain methods and transform domain methods. Pixel domain methods range from basic neighborhood filters, e.g., local Gaussian filtering, to the more powerful self-similarity based methods, the first of these is the well-known non-local means algorithm. Additionally the variational methods can be counted as pixel domain methods, the most known thereof is the total variation denoising. Transform-domain methods differ mainly in their decomposition method. Fourier methods are not best suited for denoising, as they do not provide any local information in the coefficients. This is either solved by using a DCT in a local window, or by using wavelet transformations, which provide both frequency and spatial information.

The most successful methods nowadays build on the advantages of both domains: transform domain and pixel domain. The method BM3D, for example, exploits the idea of self-similarity in pixel domain, and then applies DCT-domain filtering.

As published denoising algorithms are far too numerous to be discussed entirely in this chapter, one or two algorithms out of each group are selected. The first method, presented in Sec. 2.3.1, is an efficient low-cost denoising method, a simple neighborhood filter. The concept of self-similarity and the denoising method non-local means are presented in Sec. 2.3.2. Sec. 2.3.3 discusses DCT-based denoising methods: the state-of-the-art algorithm BM3D, which combines the self-similarity concept and DCT thresholding, and a basic sliding-window DCT denoising method. The total variation method and a new adaptation for color images is presented in Sec. 2.3.4. Classical wavelet denoising is discussed in Sec. 3.2: two algorithms are selected: the state-of-the-art denoising method BLS-GSM and a more simple approach, the bivariate denoising. The last denoising method is the sparse dictionary based denoising discussed in Sec. 2.3.6. Sec. 2.3.7 finally discusses joint denoising and debayering. Performing both tasks together seems very attractive to save resources. However, we will also discuss the drawbacks.

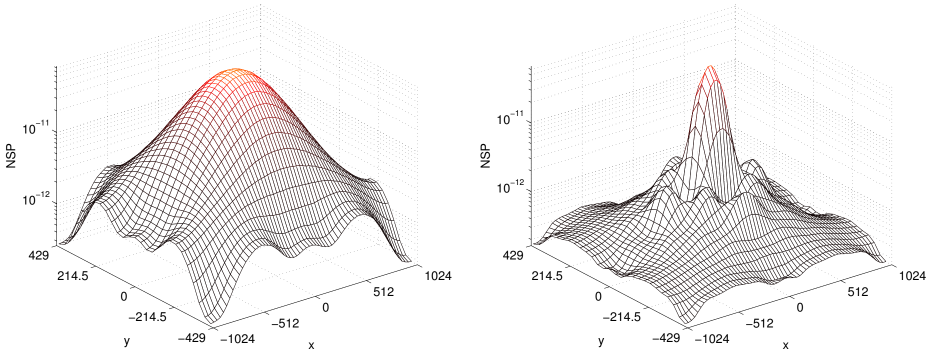
A representative set of test sequences is used for evaluating denoising methods in the following. The test set includes the computer-generated city sequence, described in the last section, and four real camera sequences that were captured with the ARRI Alexa camera. The real camera sequences are averaged over 40 frames to obtain a reference image. The reference images of these five sequences are shown in Fig.2.24.

2.3.1 Basic neighborhood filters

The most simple filters are moving-average filters, which provide efficient and low-cost denoising, but the clear disadvantage is loss of detail and sharpness. Denoising using smoothing filters, e.g., a simple Gaussian filter, still seems to draw some interest [118] for the case of scarce computational resources or time constraints in real-time applications. However, the denoising effectiveness is strongly related with sharpness loss and therefore edge-preserving filters are preferable for a better trade-off between computational cost and visual quality. One well-known and still simple solution to avoid blurring of edges while reaching high noise reduction is the so-called sigma filter, which preselects the pixels for averaging adaptively. Lee studied the well-known filter in 1983 [68].

The sigma filter selects all pixels that are similar to the pixel to denoise in a limited spatial neighborhood. The selection is based on a threshold, which is usually the noise standard deviation σ multiplied with a factor k that regulates denoising strength. The sigma filter results are very natural appearing images; compared to other filters no oversmoothing or contrast loss occurs. However, the tails of the noise distribution remain unchanged, which visually leads to single noise pixels protruding from the surrounding denoised image data, in particular when the noise level has been high.

As the sigma filter operates in a spatial window, the highest noise reduction is obtained when all the pixels in the window are used for averaging. All frequencies lower than the window size remain unchanged. The remaining low-frequency noise in the images leads to flickering if the filter is used for denoising digital video data. The effect is measured using an image sequence of grey images with camera noise. Fig. 2.25a shows the original frequency distribution of the noise. The spectrum of the denoising result in Fig. 2.25b is very different in its distribution over the



(a) noisy

(b) denoised

Figure 2.25: (a) shows the noise power spectrum of a sequence with real camera noise and (b) shows the noise power spectrum after denoising with a spatial filtering (sigma filter with 11×11 neighborhood and $k = 2.5$).

frequencies. Overall, it is lower because the noise is reduced, but the problematic part is the frequency distribution: the low frequencies are not reduced because of the limited spatial window, whereas the high frequencies are clearly suppressed. The low spatial frequencies, however, are very disturbing in video data, as they cause disturbing temporal flickering.

The sigma filter serves us as a basic example for discussing the two major trade-offs in denoising: the trade-off between noise reduction and visual quality and the tradeoff between denoising effectiveness and computational resources. Choosing a high parameter k leads to low-frequency flickering that degrades the visual quality, but it also leads to high noise reduction. Thus, a good tradeoff must be found. To achieve high noise reduction and high visual quality the spatial support should be large. This, however, requires significant buffer size in a stream based video pipeline. This trade-off between denoising effectiveness and computational resources hence mainly depends on the memory resources.

Note that the problem of remaining low-frequency noise does not only occur using the sigma filter; the temporal flickering of spatially low-frequency noise in video data appears also for other spatial denoising methods with limited spatial support.

The problem can be avoided using a temporal filter. Temporal filtering can reduce noise without sharpness loss, which seems very attractive in times of the race of resolution increase up to 4K, 6K and 8K. Regarding computational resources, one drawback is that temporal filtering leads to high memory requirements compared to a spatial filter, which only needs a very limited support window. However, if a frame buffer is available it is computationally efficient.

The sigma filter is a good choice for temporal denoising, as it can be applied temporally in almost the same way as spatially. It takes into account past and/or future values instead of the spatially neighboring values, selects all the similar pixels

Table 2.4: Sigma filter PSNR results [dB]: the spatial filtering uses an 11×11 window, the temporal the past frame, which was already denoised, both results with $k = 2.5$. The PSNR of the temporal version is the mean PSNR over 8 frames.

Method	domain	ball pool	candy	city	street	arch
Noisy		30.86	30.14	35.91	37.04	32.68
Spatial, camera σ.	bayer	**32.36**	**30.97**	37.13	38.12	34.39
Temporal, camera σ	bayer	31.90	30.52	36.08	37.84	32.83
Spatial, const. σ	linear	31.52	29.00	**37.81**	37.64	33.44
	LogC	31.00	29.81	36.77	38.60	35.82
	mon	30.82	29.93	37.01	**38.97**	**35.85**
Temporal, const. σ	linear	29.51	27.65	36.04	36.54	32.01
	LogC	30.61	29.79	36.04	38.03	34.89
	mon	30.65	29.81	36.06	38.03	34.58

according to the $k\sigma$ threshold, and reduces the noise by averaging. In the case of temporal denoising, the selection of pixels does not avoid edge blurring, but serves as a basic motion detection method: pixels deviating more than the threshold are likely to belong to a different object and thus have to be excluded from averaging to avoid motion blur. One may argue that in this case a temporal correlation is introduced, which is true. But as long as the temporal support is low, this does not disturb the human vision too much.

To limit the memory requirement we propose a temporal filter that is applied on the past denoised frame, so it requires only one frame in the memory. Thereby we implicitly provide a larger temporal support, which contributes with diminishing influence, hence keeping the temporal correlation low enough to avoid disturbance. The temporal support, however, still provides a much lower amount of potentially similar pixels; hence the noise reduction will be lower than in spatial filtering.

The described variants are applied to five sequences: the computer-generated city sequence, which allows calculating objective metrics based on the available full RGB reference, and four sequences captured with the Alexa camera with static image content, which can also be used for metric calculation, as the mean of all images belonging to one sequence can serve as reference.

Tab. 2.4 shows the PSNR of the different described sigma filter variants. The variants differ in supporting pixel values (spatial/temporal), domain in which the filter was applied (linear/LogC/monitor), and constant or signal-dependent noise standard deviation.

The signal-dependent noise standard deviation is estimated based on the noisy pixel value and the measurements presented in Sec. 2.2.1. This is a fairly good estimation, as the variance does not change with a steep slope and thus it can be assumed to be equal to the noise variance of the true pixel value. For the RGB data, a constant threshold parameter based on the measured noise standard deviation in

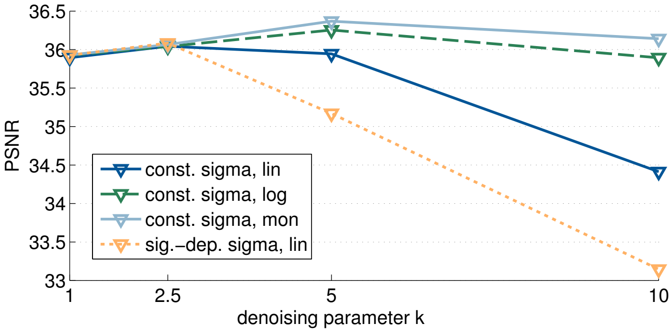


Figure 2.26: Results of denoising the City sequence using a temporal Sigma-Filter. The parameter k is multiplied with the noise standard deviation σ, to regulate denoising strength. In linear domain (lin) the signal-dependent camera noise was used. Additionally a constant sigma value was used for all three domains. The filtering in the latter case is homogenous.

the complete image is used. This is chosen because most filters from literature assume a constant noise standard deviation, and because no simple model exist for the noise after color processing. As difference images have shown that the mean reference image deviates partially from the true value – due to small movements or changing lighting conditions – the overall noise standard deviation calculated using the mean of the squared difference of the noisy image I_{noisy} to the reference image I_{ref} has not been robust enough; causing a clearly too high standard deviation, this effect has led to extremely oversmoothed denoising results. Therefore, the standard deviation of the noisy images is calculated using the median instead of the mean.

$$\sigma_{const} = \sqrt{med\left((I_{noisy} - I_{ref})^2\right)} \tag{2.16}$$

This provides more robust estimates and reliable denoising strength.

The parameter k cannot be directly optimized for each sequence, as higher k always lead to higher PSNR for static sequences but also to motion artifacts in non-static sequences. It therefore was fixed considering visual quality, PSNR values and the theoretical recommendation by Lee [68]. The PSNR values of the city image are shown in Fig. 2.26. The PSNR is clearly rising for all different sigma filter variants up to $k = 2.5$. Higher values for k gave better results in monitor and LogC domain, but the improvement is not large. For linear data, the PSNR clearly drops for k-values higher than 2.5. Additionally, higher k can lead to motion artifacts and thus the parameter was fixed to $k = 2.5$ for all sequences. This parameter is close to the parameter proposed by Lee [68].

The PSNR values in Tab. 2.4 show, as expected, the lower noise reduction of the temporal filter, which is due to the lower support pixel values. The spatial filter was applied with a 11 × 11 window, which provides 120 neighboring values, and the temporal filter was applied using the past denoised frame, which provides one pixel value. However, as the past pixel value is already denoised and very likely to belong to the same true signal value, the PSNR of the temporal filter is only around one dB below the spatial filter result. The temporal solution hence provides a good trade-off between computational complexity and quality of the denoising result.

Whereas the filtering based on the camera noise variance must be applied on the Bayer data, the filtering based on the constant σ allows to compare filtering effectiveness in linear, logarithmic, or final display domain data. The constant σ, however, does not provide comparable results to the signal-dependent σ on the ball pool and candy sequence. Both of these sequences have content with very saturated colors, and thus the signal differences are large. As significantly different signal values also mean that the noise variance is significantly different, this data requires a signal-dependent denoising parameter. City, street and arch sequences show large signal differences only in very small regions around highlights, which almost do not influence the overall PSNR. Therefore, relatively high denoising quality can be achieved with a constant σ on these sequences.

In terms of PSNR no image domain can be stated to be preferred in general. The temporal filtering is most effective in LogC or video domain, depending on the sequence. For the spatial filtering, however, the linear domain leads to the highest PSNR for the ball pool and the city sequence, whereas for the other three sequences the highest PSNR is obtained by denoising in the video domain.

The sigma filter provides high-quality denoising results given its low computational cost. The PSNR can be increased up to 3 dB and the noise is clearly reduced compared to the original sequences. For $k < 2.5$ no motion artifacts could be observed in sequences containing motion. The spatial filter retains edges, but it can smooth details and lead to unpleasant flickering due to spatially low-frequency noise.

2.3.2 Self-similarity

The previously described filter approach is based on averaging the values of similar pixels. The similarity is determined based on the pixel values: when the difference is not larger than the noise standard deviation the pixel is taken into account for averaging. This very basic decision is extended by a concept called “self-similarity”, which is exploited in the non-local means (NLM) algorithm, presented by Buades in [20]. The NLM includes not only the pixel value itself, but also each pixels neighborhood for the decision if a pixel is similar. This self-similarity concept assumes significant redundancy in natural images: most image structure is not unique in natural images, but similar image content can be found within a natural image. Of course, this concept can, same as the sigma filter described in the last section, easily be extended to temporal filtering, by just extending the support to the preceding and/or subsequent frames.

The NLM algorithm consists of three steps: block matching to select similar blocks within the image, weighted averaging to obtain an estimate for each block,

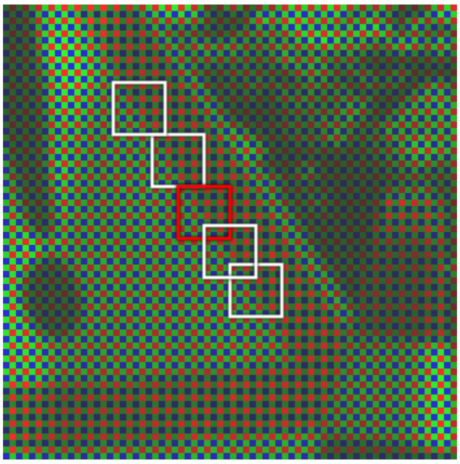


Figure 2.27: Matching blocks in NLM denoising on Bayer data: when the red block is denoised, pixels from other blocks with similar structure are taken into account.

and fusion of the estimates, which is necessary because the blocks are overlapping. The resulting estimate y depends on the noisy image z in the neighborhood $\mathcal{N}$ around the pixel i.

$$y(i) = \frac{\sum\limits_{j \in \mathcal{N}(i)} w(i,j) \cdot z(j)}{\sum\limits_{j \in \mathcal{N}(i)} w(i,j)} \tag{2.17}$$

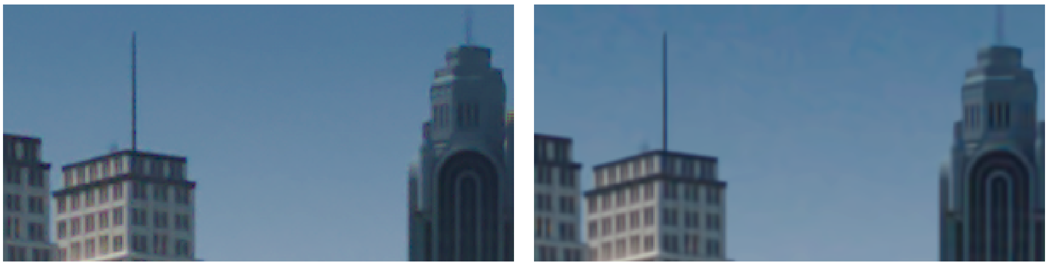
The pixels weights w depend on the similarity between the patch $B(i)$ around pixel i and $B(j)$ around pixel j: larger Euclidean distance leads to lower weights.

$$w(i,j) = \exp\left(-\frac{||B(i) - B(j)||^2}{h}\right). \tag{2.18}$$

The filter parameter h is usually set empirically based on test data. As mentioned above, $B(i)$ denotes a patch around the pixel i and $\mathcal{N}(i)$ the search area around the pixel i. Hence, additional parameters are the block size of B and the size of the neighborhood $\mathcal{N}$.

Adaption for Bayer data The algorithm can very easily be adapted and used on Bayer data: we just restrict the step in the block matching procedure such that only blocks with the same pattern are taken into account. Fig. 2.27 shows an example of the block matching on Bayer data. In the example, the block in the red square is denoised using the blocks in the white squares. Only blocks with the same pattern – here, red Bayer pixel in the upper left corner– are taken into account.

Results for the NLM on Bayer data are shown in Fig. 2.28. Whereas the original NLM often leads to oversmoothed results, the Bayer NLM keeps the image quite



(a) NLM Bayer

(b) BM3D

Figure 2.28: NLM denoising on Bayer data for the city sequence. The sky, a smooth area, is better denoised than in BM3D. BM3D is applied on color RGB data with spatially correlated noise. NLM is applied on Bayer data with 7×7 blocks, and a search window of 37 pixels.

sharp but denoises well the smooth area – the sky – in the picture. However, color artifacts are visible in the small windows of the left house and the right side of the right tower. The result depends on the chosen parameters; based on optimizing the Bayer NLM search window $\mathcal{N}$ from width 5 to 37 with the block widths 3, 7, and 11, the highest PSNR is 39.21 dB, based on a neighborhood width of 37 pixels width and a block size of 7×7.

The concept of self-similarity is exploited quite successfully in the BM3D algorithm, a state-of-the-art algorithm that leads to almost optimal results – in terms of MSE [27]. The BM3D uses similar blocks within an image, as NLM does, but on these blocks a more sophisticated denoising procedure is applied: the blocks are grouped and denoised with a 2D-DCT and a Haar wavelet transform in the third dimension. On the resulting coefficients, denoising is realized using hard thresholding. In a second step, the basic estimate from the first step is filtered again, except that Wiener filtering is used instead of hard thresholding.

The BM3D, applied on the processed monitor data, leaves a quite significant amount of noise in the smooth area of the image, but retains well the structure. Although the algorithm yields good PSNR results (41.06 dB on the city image), the results are often oversmoothed but homogenous regions remain noisy. The remaining noise is, as often after spatial denoising, low-frequency noise, which is known to be very disturbing, especially in video sequences. Fig. 2.28 compares both algorithms. Despite its lower complexity, the Bayer NLM result looks better in the sky, but compared to BM3D the contrast in the building on the right is reduced and some colored artifacts are visible.

BM3D is extended to video (VBM3D), which just uses blocks not only from one image but also from other frames. However, this does leave the problem of low frequency noise. Although Buades in [21] states: "In all of these movie denoising algorithms the neighborhood J(i) picks a single pixel per frame. One of our objectives is to prove that this restriction is actually counterproductive. In fact the performance of movie denoising filters improves significantly by forgetting about trajectories and using all similar pixels in space-time, no matter how many are picked per frame. For this reason the NL-means filter treats movies as a union of images, rather than an image sequence.", Liu and Freeman disagree [75]. Especially in smooth areas block matching methods are not the best choice: in smooth areas the blocks are matched by the most similar noise pattern without a temporal consistency analysis. Perhaps no sophisticated motion estimation is necessary, but methods should include some regularization that works towards a temporally stable solution and thus reduces flickering. Future research should concentrate on avoiding spatially low frequency noise, which is very disturbing in videos.

Although temporal consistency is mentioned to be included in the VBM4D algorithm, presented by Maggioni et al. in [80], the results using the Matlab implementation provided by the authors on the five selected camera noise sequences are not better than BM3D: not more noise is removed. The denoising strength is low, and hence, also the overall PSNR is low with 34.88 dB (mean over 5 sequences) and for the city sequence 38.11 dB. However, this may be due to suboptimal parameter settings. The algorithm has different parameters, and the parameters provided by the authors may be not optimized for camera data, but, as usual, for grey sequences with artificial AWGN. The selection of the optimal parameters is beyond the scope of this dissertation.

2.3.3 DCT based methods

The DCT is a well-known transformation that is used in most popular image compression methods, as it provides a sparse representation of the image signal and is relatively easy to implement. Usually, it is applied patch-wise, e.g., with 8 × 8 or 16 × 16 patches. The transformed patches can be denoised using thresholding techniques. To avoid blocking artifacts, the DCT must be applied in a sliding window manner with overlapping patches. Patch estimates are the obtained by separate thresholding of the patches and the patch estimates are finally merged by averaging. The DCT denoising can be applied in RGB color space or in a decorrelated luma-chroma color space; the latter one usually provides better results.

Image examples of a DCT thresholding are shown in Fig 2.29, the PSNR of the candy sequence crop is 31.95 dB. The DCT can well denoise the high frequency noise, but low frequency noise remains in the image. Whereas this is not a big problem in still image denoising, the low-frequency noise leads to flickering in denoised video sequences. To reach optimal PSNR, the denoising strength must be increased until the image is clearly oversmoothed and small artifacts (overshooting) at edges appear. Overshooting refers to an output that takes on higher values than the input, and it comes along with undershooting, which describes an output value lower than the input.

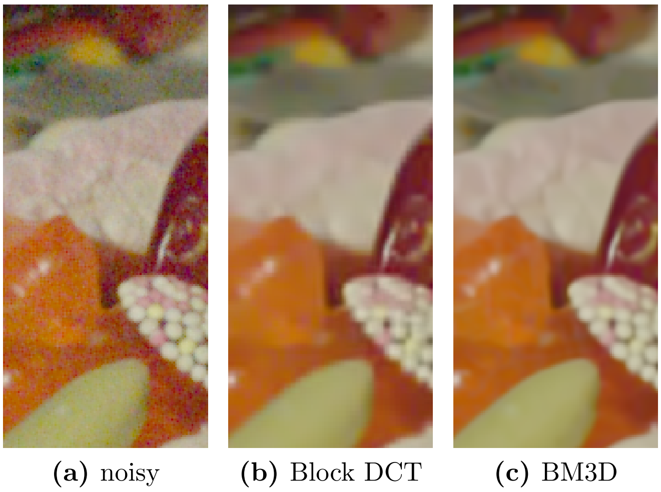


Figure 2.29: Image crops from the candy sequence (PSNR-optimized): denoising with overlapping blocks and hard thresholding in the DCT domain is shown on the left, denoising using the state-of-the-art method BM3D on the right.

The well-known state-of-the-art denoising method, building on a DCT transformation, is BM3D [33], described in the last section. This method provides good results in terms of PSNR (candy sequence 33.09 dB). The result of BM3D applied to the candy sequence crop is shown in Fig. 2.29c. Although the structure is better retained than with the basic DCT thresholding, the PSNR-optimal result is oversmoothed and low-frequency noise remains. When the denoising strength is set to reach PSNR-optimal results, the parameter σ is ten times the measured noise standard deviation. The strong denoising leads to oversmoothed results. Compared to the simple DCT thresholding, however, more details are kept (e.g., in the rose part in the middle) and no artifacts at edges appear.

2.3.4 Total Variation

Variational methods for image reconstruction are widely used due to their flexibility and the applicability to a wide range of problems in image processing. The typical problem formulation is a minimization problem based on a data term and a regularization term imposing some kind of smoothness or prior information [16, 23, 79]. Many denoising methods mentioned in this dissertation could also be formulated using this type of description.

For denoising, the problem formulation consists of a data term that keeps the denoising result close to the noisy image and a regularization term that penalizes the noise and hence is responsible for the denoising effect. A well-known regularization method has been presented by Rudin et al. in [104]; it is the so-called total variation (TV).

$$\mathrm{TV}(u) = \|\sqrt{(D_x u)^2 + (D_y u)^2}\| \tag{2.19}$$

$D_x u$ and $D_y u$ are the derivatives of the image in x- and y-direction respectively. The total variation is a measure of the local activity of a signal; it is high when the sum of the absolute gradients is high. As natural images are smooth in large image parts, the total variation is low. When noise is added, the sum of absolute gradients rises and thus the total variation increases.

The method is designed to remove noise while keeping the edges in an image sharp and hence aims to avoid oversmoothing. It is very effective in retaining edges, however, it unfortunately also reduces contrast, especially in small details. To avoid this contrast loss, the Bregman iteration method was proposed [17,94], which iteratively reconstructs the contrast, starting from a TV estimate.

Most of the available methods were proposed for grayscale images, in particular Bregman iterations have been investigated on grayscale images or single channels of a color image. Different TV variants have been proposed for color data, e.g., Vectorial TV [18,44,108]. The idea of our color Bregman TV method, proposed in [9], is to use the Bregman distance regularization in an iterative setting to make use of the coinciding edge sets in the different channels.

Color Bregman TV

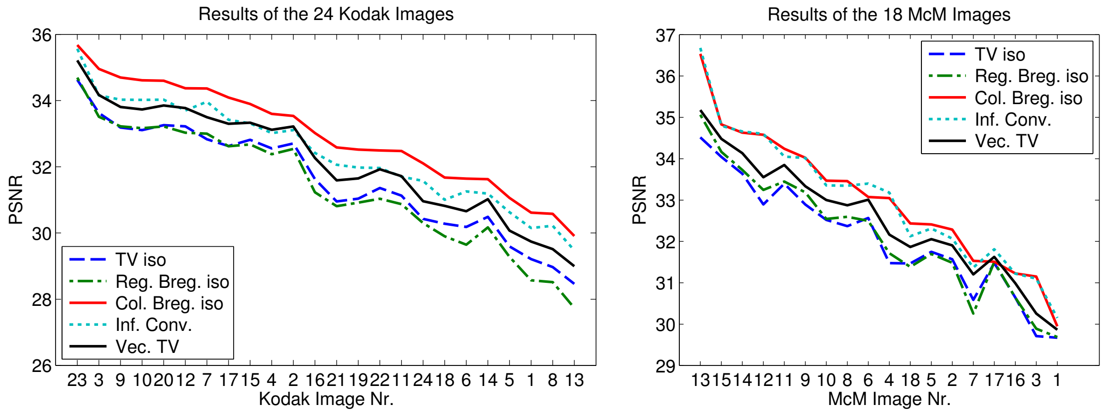
The motivation for our novel iterative procedure for multichannel image reconstruction, published in [9], is the co-appearance of edges and image structures in multiple channels of natural images. The proposed color Bregman method aims at a solution with reasonably small total variation. Additionally, it encourages different color channels to have a common edge set by requiring the channels to share a common subgradient of the TV functional 2.19. This way, no assumption has to be made on the size of the jumps across the edges in each channel and better denoising efficiency with good detail and contrast preservation can be achieved.

As the constraint of a common subgradient is difficult to impose directly, we use the Bregman distances between color channels in an iterative procedure to measure the differences between edge sets of different channels. We determine each iterate by regularizing each channel with a weighted linear combination of Bregman distances to all other image channels from the previous iteration. In this sense, we generalize the Bregman iteration proposed by Osher et al. in [94] to multichannel images.

A common subgradient of different image channels forces the edges to – loosely speaking – ‘go into the same direction’. We also consider the case of different channels sharing an edge set but not (necessarily) the edge direction. Mathematically, we use the infimal convolution of Bregman distances to achieve the independence of the edge direction. Whereas a more detailed mathematical description and convergence analysis of the proposed scheme can be found in our paper [9], this section presents

Table 2.5: Mean PSNR results [dB] of the 24 Kodak images of the 18 McM images with AWGN ($\sigma = 0.05$).

Method	PSNR Kodak	PSNR McM
TV	31.59	32.07
Channel-by-channel Bregman iteration	31.38	32.15
Color Bregman iteration	**32.95**	**33.02**
Infimal convolution Bregman iteration	32.41	33.01
Vectorial TV	32.16	32.52



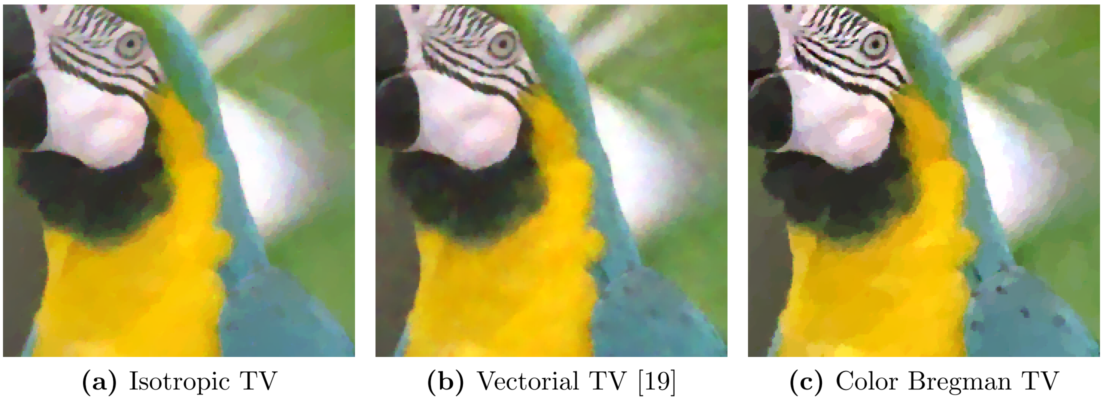
(a) (b)

Figure 2.30: PSNR results for the 24 images of the Kodak image set on the left and for the McM data set on the right, sorted in descending order ($\sigma = 0.05$).

denoising results based on two different usual data sets and on more realistic camera data.

For our numerical experiments, we used the 24 images of the Kodak image data set, which is usually considered to be a representative set of color images. We added Gaussian noise with a standard deviation of $\sigma = 0.05$ (for images on a scale from 0 to 1) to obtain noisy color images and applied a channel-by-channel TV, regular Bregman iteration, infimal convolution Bregman iteration and color Bregman iteration to the noisy images and additionally compare our results to the vectorial total variation (VTV) approach proposed by Bresson and Chan in [19]. The parameter μ for the Bregman iteration methods and the parameter λ for the VTV method were optimized using the first five images of each data set and then these parameters were used for denoising the complete image data set. For the color Bregman TV iteration, we set all weights to one third, $w_{i,j} = 1/3$ since this choice gave the best reconstruction quality in our numerical denoising experiments.

The mean PSNR values over all 24 Kodak images are shown in Tab. 2.5. As we can see, color Bregman iteration yields the highest and infimal convolution Bregman



(a) Isotropic TV (b) Vectorial TV [19] (c) Color Bregman TV

Figure 2.31: Crop of the Kodak image no. 23.

iteration the second highest PSNR value. The mean PSNR using color Bregman iteration is about 1.35 dB higher than the PSNR using classical TV. In comparison to VTV, the infimal convolution approach yields a small improvement and color Bregman iteration show a PSNR gain of around 0.8 dB. Fig. 2.30a shows the PSNR results of all methods on all 24 Kodak images as a line plot. For the sake of readability the results are sorted in descending order based on the color Bregman PSNR value. As we can see, the color Bregman iteration results show the highest PSNR for all the 24 images. The infimal convolution approach outperforms VTV in most cases but remains clearly behind the color Bregman iteration results. Thus, for our test data, the stricter assumption of the edges pointing into the same direction seems to help much more with the suppression of the noise than it harms at parts where the assumption might be violated.

We additionally used the McMaster (McM) data set. Similarly to the Kodak data results, the mean PSNR values in Tab. 2.5 show that color Bregman iteration yields the best results in terms of PSNR. Interestingly, on this dataset the infimal convolution iteration yields almost equally good results. This may be due to the extremely strong colors in the images from the McM data set, which may lead to a large amount of edges pointing in different directions. Fig. 2.30b shows that the color Bregman iteration outperforms VTV on all images with one exception: image number 17, which shows a highly saturated image of a green and red flower. As the edges mostly point into different directions, infimal convolution shows the best result on this image.

A visual comparison of the results obtained on Kodak image 23 (see Fig. 2.31) shows the advantage of obtaining higher contrast images with color Bregman iteration. The contrast and detail recovery is highest in the color Bregman results. Notice that a higher noise level of $\sigma = 0.15$ was chosen in this example to better show the differences of the results of the methods. Fig. 2.31a shows the results of classical channel-by-channel TV, Fig. 2.31b the result of vectorial TV, and Fig. 2.31c

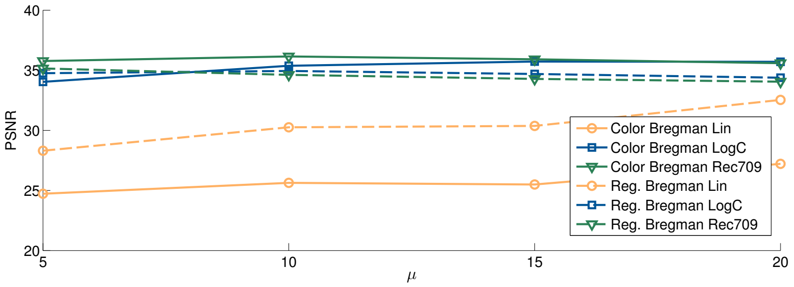
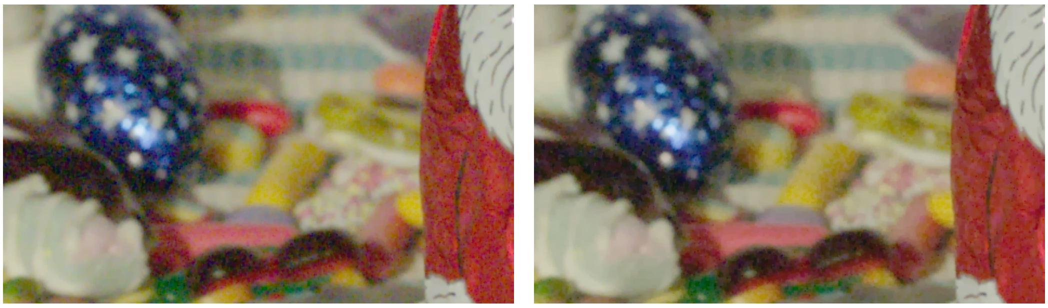


Figure 2.32: The mean PSNR of the five sequences (ball pool, candy, city, arch, street) for the proposed color Bregman method and the regular channelwise Bregman method is plotted over the denoising parameter μ.



(a) Regular Bregman

(b) Color Bregman

Figure 2.33: Image examples of the color Bregman method and the regular Bregman method, applied channelwise.

the proposed color Bregman method. Around the eye of the bird we observe higher contrast in the color Bregman results compared to the TV results. The black color in the circle around the eye can be recovered completely using the color Bregman method. The color Bregman method is the only method that brings back details (dark spots) on the blue wing of the bird.

The color Bregman method is additionally applied to real camera data in the linear, LogC, and display domain. On the five sequences of real camera data and the city sequence, the achievable gain in PSNR depends on the domain. Fig. 2.32 shows the mean PSNR over all five sequences plotted over the denoising parameter μ. The iteration number was optimized to yield highest PSNR for each μ. In display domain, the mean PSNR is 1 dB higher for color Bregman, compared to regular Bregman. In LogC domain, the difference is slightly smaller (0.7 dB), and in linear domain, the regular Bregman method applied channelwise achieves clearly better PSNR results. Visual inspection, however, shows that the moderately applied color Bregman method yields a stronger suppression of colored noise than the regular Bregman result (see Fig. 2.33), and hence seems to provide high visual quality.

Table 2.6: PSNR results in dB of wavelet-based denoising methods: bivariate denoising and BLS-GSM is applied on full-color image sequences in linear, LogC, and display domain.

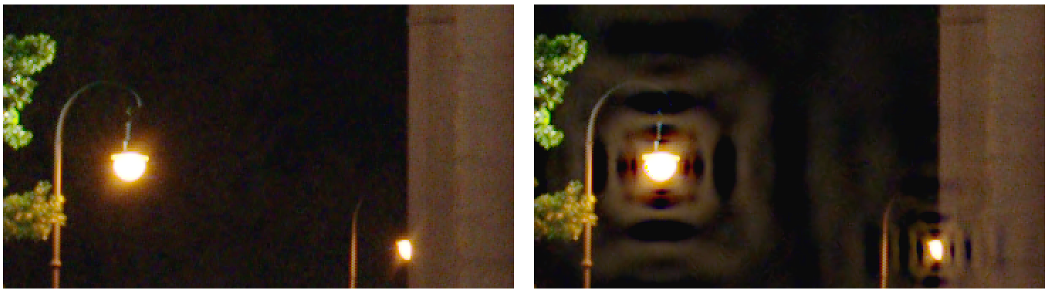
Method	domain	ball pool	candy	city	street	arch
Noisy		30.86	30.14	35.91	37.04	32.68
Bivariate orig. (YCbCr)	linear	29.84	28.05	36.58	37.07	32.45
	LogC	31.03	29.82	36.39	38.04	34.70
	mon	30.94	29.84	36.45	38.00	34.61
Bivariate orig. (RGB)	linear	30.43	28.82	36.94	37.41	32.93
	LogC	30.66	29.81	36.50	38.03	34.75
	mon	30.74	29.83	36.48	38.02	34.64
BLS-GSM (σ_{const})	linear	30.68	29.23	36.80	31.58	29.94
	LogC	**33.81**	31.41	**38.46**	38.33	34.73
	mon	33.48	**32.22**	37.99	**38.53**	**35.80**

2.3.5 Wavelet Denoising

Denoising images usually works well in a domain providing a sparse representation. A common tool for that is a wavelet decomposition, which transforms the image pixels into wavelet coefficients. The term wavelet decomposition, however, does not correspond to one single filter. Instead, multiple different decompositions are possible, differing in their intended application and computational cost. The denoising procedure applied to the wavelet coefficients can be equally various: from simple thresholding techniques up to expensive Bayesian estimation methods. A relatively simple and still very effective method is the bivariate thresholding by Sendur et al. [111]. Being computationally relatively simple, it exploits wavelet coefficient dependencies and thereby reaches significantly better results than thresholding techniques that assume independency of the wavelet coefficients.

The bivariate shrinkage is based on a model that captures the dependence between a wavelet coefficient and its parent. Sendur and Selesnick in [111] derived from this model a simple non-linear shrinkage function to reduce the noise. The shrinkage function depends on the coefficient and its parent, and it includes a threshold calculated based on a local estimation of the signal standard deviation and an estimation of the noise standard deviation.

Bivariate denoising using a simple Haar transformation is the easiest approach implementable with very low implementation cost. Tab. 2.6 provides the PSNR values for the five test sequences. Given the extremely simple Haar wavelet basis, an increase the PSNR about up to 2 dB is considerable. However, not always a PSNR gain can be achieved. The visual results show that at edges blocking artifacts can occur when using the method with this simple transformation, but overall the results show high visual quality. Fig. 2.34a shows an image crop from the arch sequence. As



(a) Bivariate (b) BLS-GSM

Figure 2.34: The image crops of the arch sequence show denoising results on linear data. The bivariate shrinkage (a) does not reduce the noise much, but no artifacts appear; BLS-GSM denoising (b) leads to strong artifacts when applied on high dynamic range sequences.

the method is quite efficient, Chapter 3 of this dissertation proposes a modification for camera data.

A more sophisticated state-of-the-art wavelet denoising method is the BLS-GSM algorithm, proposed by Portilla et al. [101]. To reach highest performance, it bases on a more sophisticated model, which additionally accounts for the local correlation between coefficients. Using an overcomplete multiscale oriented basis, it reaches high-quality denoising results. The method reaches higher PSNR values than the bivariate method on LogC and on display domain data. When applied to linear camera data, however, the method shows strong artifacts. On the high dynamic range scenes captured at night (street and arch) strong ringing artifacts occur around highlights. Ringing describes the effect when the output signal oscillates around a sharp transition in the input, e.g., an edge between very bright and very dark image content. Fig.2.34b shows the artifacts.

2.3.6 Sparse Representation Denoising

Sparse representation of image data has been successfully applied to various problems of image reconstruction such as denoising, demosaicking and super-resolution with impressive results [14, 81, 82, 127]. Based on the assumption that image signals admit a sparse decomposition over a redundant dictionary, efficient methods have been developed by approximating a noisy image patch y_i by a sparse linear combination of elements from the dictionary.

In literature, AWGN is used to demonstrate denoising performance on grayscale data by Aharon et al. in [14] and on readily processed color image data by Mairal et al. in [82]. Regarding real camera data, the method has been applied to some low-dynamic-range camera raw data in a following publication by Mairal [81], but without adapting it to the noise characteristics of the camera. In this section the method is shortly explained and adapted to the signal-dependent noise characteristic described in Sec. 2.2.1. The method can easily be extended to joint debayering and denoising, which will be discussed in Sec. 2.3.7.

Sparse representation

The method of sparse representation approximates a signal $y \in \mathbb{R}^n$ by a linear combination of so-called atoms from a dictionary $D \in \mathbb{R}^{n \times k}$. Aharon and Elad used the K-SVD method [36, 37], to train an over-complete dictionary D from a set of l training data $Y = [y^1, ...y^l]$. Describing the training method is beyond the scope of this dissertation, for details refer to Aharon et al. [14]. Aharon showed impressive results in denoising grayscale images using the K-SVD algorithm to train a dictionary on the noisy image itself. Therefore the image is divided in l overlapping patches of size n. Each image patch is processed separately and the final estimate is achieved by averaging the reconstructed patches. As this approach, however, requires significant memory resources, we use a pretrained dictionary in the following.

The concept of sparse representation assumes that a signal y can be approximated by $D\alpha$ with the representation vector α containing only a small number of non-zero coefficients. To find the sparse representation, α is determined solving the following minimization problem,

$$\hat{\alpha} = \arg\min_{\alpha_i} \|\alpha_i\|_p \quad \text{s.t.} \quad \|y_i - D\alpha_i\|_2^2 \leq \epsilon \tag{2.20}$$

with p typically $\{0, 1\}$. As there is no unique solution to $\|y_i - D\alpha_i\|_2^2 \leq \epsilon$ since $k > n$ for a redundant dictionary, the regularization term $\min_{\alpha_i} \|\alpha_i\|_p$ is needed. For $p = 0$ equation Eq. 2.20 counts the non-zero elements in α. The solution is approximated using greedy algorithms like the Orthogonal Matching Pursuit (OMP) [120].

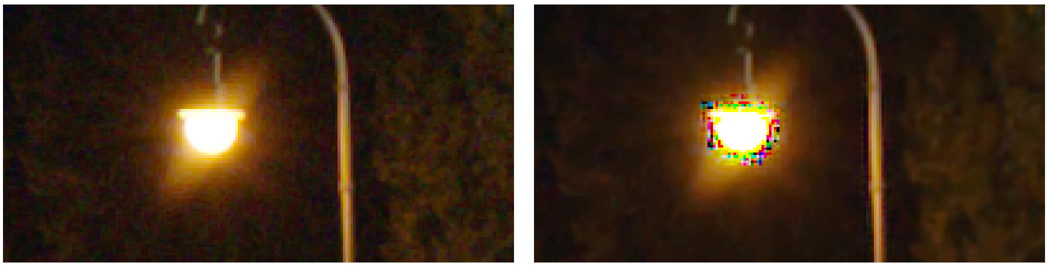
The reconstruction $D\alpha$ is assumed to tend towards the noise-free image and the allowed reconstruction error e_i^2 should correspond to the noise because noise is a high frequency and random disturbance to the image, which requires many atoms to be reconstructed.

We choose the threshold ϵ for the OMP based on σ, patch size n, and a constant factor c. The constant c is chosen aiming best PSNR. To prevent the OMP from an over-adaptation of the patch we also choose a maximum sparsity of $m = 12$ atoms. The dictionary D is a global dictionary trained on a fixed image set using the K-SVD method with 15 Iterations and $500,000$ training patches. The patches have a size of $7 \times 7 \times 3$ pixels.

As the noise in camera data is signal-dependent, we take into account the signal dependency, which is known for the Bayer domain and is assumed to be similar after debayering. We rely on a standard deviation σ estimated individually for each noisy pixel. There are two approaches presented for taking into account the value σ for the denoising. Weighting each pixel individual based on its σ-value – already established as "extension to nonuniform noise" by Mairal [82] – or weighting

Table 2.7: PSNR [dB] of the sparse-dictionary-based denoising method and the adaptation for signal-dependent noise, applied on the linear city sequence after debayering (linear) and on display domain data.

denoising domain	linear	display
no modification	38.96	39.02
local OMP threshold	**38.99**	**39.28**



(a) noisy **(b)** denoised

Figure 2.35: Artifacts around highlights: for high differences of the pixel values within one patch, artifacts appear.

each patch individually – the corresponding ϵ_i is then calculated for each patch by $\epsilon_i = \sqrt{n}c\sigma_i$, based on the patch σ_i. Whereas the individual weight for each pixel lead to no improvement in PSNR, the PSNR can be increased about 0.3 dB using the local threshold ϵ for each patch; Tab. 2.7 shows the PSNR values calculated on the city image. For the local OMP threshold in display domain, a variance curve in display domain was used. Although this is not very precise and impractical in real applications, it leads to the best PSNR results. However, this improvement is extremely small and does not lead to a clear visual difference.

As mentioned above, images without gamma transformation have been used in literature. However, these images have very low dynamic range and therefore are not very different from the readily processed data. Denoising linear data with high dynamic range from extreme darkness to the highlights leads to artifacts in the final display-domain image, as illustrated in Fig. 2.35. However, a logarithmic transformation before denoising the linear data eliminates this problem, as it reduces the strong gradients around highlights.

The sparse reconstruction can also be used on Bayer data. We tested two different versions. To respect the Bayer pattern in the denoising method, in the first version, only patches with the same RGB pattern are used for training. Since the Bayer pattern repeats every second row and column, training patches are selected accordingly and denoising is applied to ever second patch. Tests show that this leads to an improvement in quality of around 3 dB in the PSNR, compared to denoising each color separately.

To denoise every patch, in a second version, one dictionary per pattern is trained. Hence, there is a dictionary to denoise every patch in the Bayer image. This improves the visual quality at edges, as well as the PSNR by about 0.5 dB.

2.3.7 Joint Denoising and Debayering

A broad range of new or newly extended methods have been proposed in the last years [25,46,52,58,63,83,86,131], to denoise Bayer sensor data directly with debayering. The more sophisticated methods operate completely in frequency domain, whereas the methods more easy to implement interpolate between pixels and estimate pixel values.

In this section, the focus is on the relatively simple methods based on local polynomial approximation (LPA) and intersection of confidence intervals (ICI) [95,96]. We additionally include a short section on sparse dictionary based denoising and debayering, which was presented in last section and can easily be applied to joint denoising and debayering.

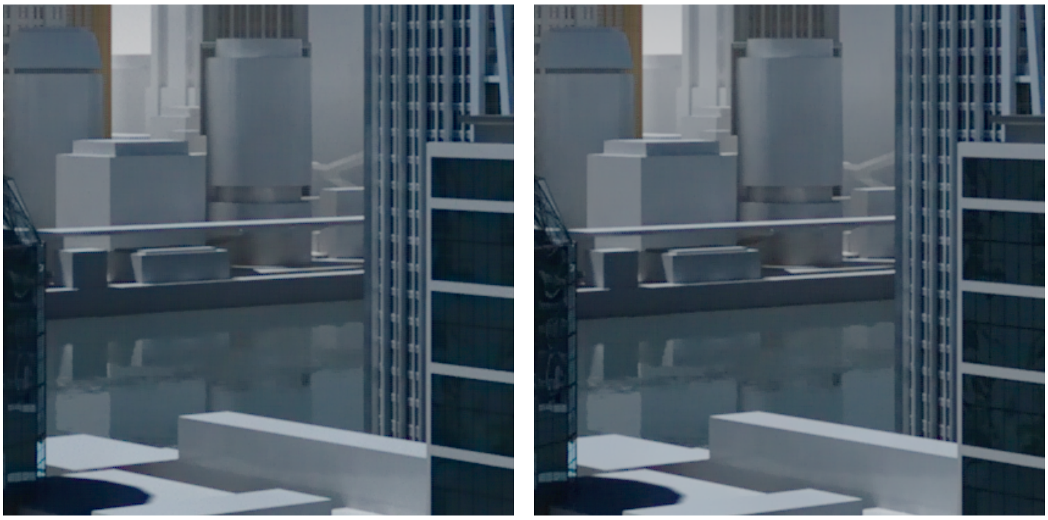
Sparse reconstruction

The sparse reconstruction described in Sec. 2.3.6 can also be used for joint debayering an denoising.

The sparse decomposition for Bayer data with respect to a full-RGB dictionary allows to reconstruct the full-color linear RGB image directly, which means the debayering step is performed jointly with the denoising. Usually, debayering algorithms interpolate the missing color values based on the Bayer data and keep the color values of pixels from the Bayer data. In contrast, sparse representation can replace the color values of the Bayer data with the reconstruction as well and thus denoises the data simultaneously. While the results of the sparse reconstruction debayering remain behind the camera debayering Ada3 for low noise, for high noise levels the joint denoising and debayering leads to approximately equal visual quality compared to the approach of debayering first and sparse reconstruction denoising as a second step – for lower computational cost as both is done in one single step. However, the sparse reconstruction requires high resources for the OMP, such that it is not adequate for a hardware implementation.

LPA-ICI

Two methods using the LPA-ICI technique were proposed by Paliy et al. [95,96] for performing denoising and debayering jointly. The first method – LPA-ICI on sums and differences (LPA-ICI S/D) [96] – uses decorrelated color channels for the LPA-ICI filtering; the filtering is performed on the differences of the green-red and green-blue channels. The second method – LPA-ICI with cross-color filters (LPA-ICI CCF) [95] – directly works on the subsampled Bayer image using cross-color filters.



(a) $\sigma = 4$
PSNR = 36.65 dB

(b) signal dependent
PSNR = 37.26 dB

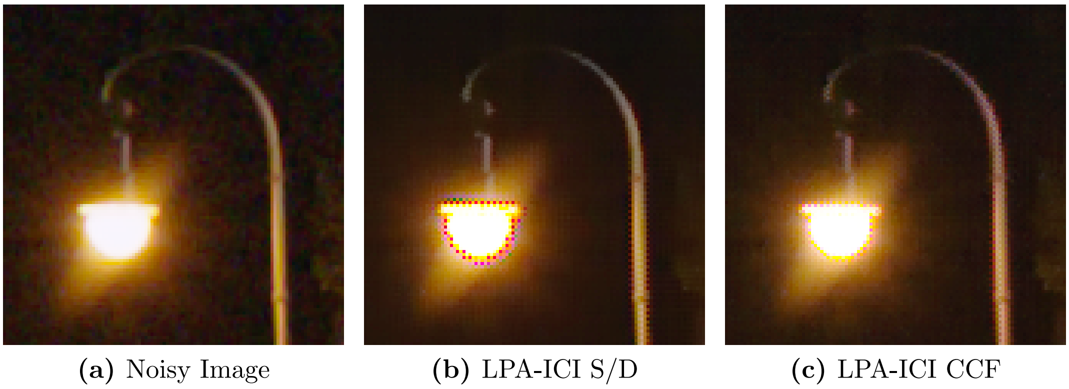
Figure 2.36: In the original LPA-ICI S/D method [96] a constant standard deviation is assumed. This does not match the real camera noise, the method can be improved by using the signal-dependent sensor noise standard deviation for each pixel.

Both methods can be improved by an adaptation to the ALEXA noise characteristic and by integrating more sophisticated debayer algorithms.

Motivated by the usual assumption that the differences of color channels are spatially slowly varying, the LPA-ICI S/D [96] is carried out on differences and sums of G-R and G-B calculated by initializing the required values not available in the Bayer data with a basic interpolation step. This corresponds to a partial debayering.

The LPA-ICI-CCF method uses cross-color filters applied directly on the Bayer data. The filters are composed of a combination of smoothing and derivative kernels [95]. The method can be separated into three steps: the first step is denoising the Bayer data and the second step interpolating the green channel. Based on the denoised R, G, B values of the Bayer data and the complete green channel, in the third step the missing R, B values are interpolated using pre-designed interpolation kernels. For performing this cross-color filtering of the Bayer image two sets of LPA kernels are needed, one for denoising and one for interpolation.

In the original LPA-ICI method [96] a constant standard deviation is assumed. This does not match the real camera noise. Both methods are implemented and the noise standard deviation is set to the noise characteristics described in Sec. 2.2.1. Fig. 2.36 shows the difference between the LPA-ICI S/D method using a constant



(a) Noisy Image (b) LPA-ICI S/D (c) LPA-ICI CCF

Figure 2.37: LPA-ICI S/D and LPA-ICI CCF color artifacts at highlights and zipper artifacts at edges.

Table 2.8: LPA-ICI with debayering in comparison to other state-of-the-art denoising and debayering methods, PSNR [dB] calculated on the city image (EI 3200).

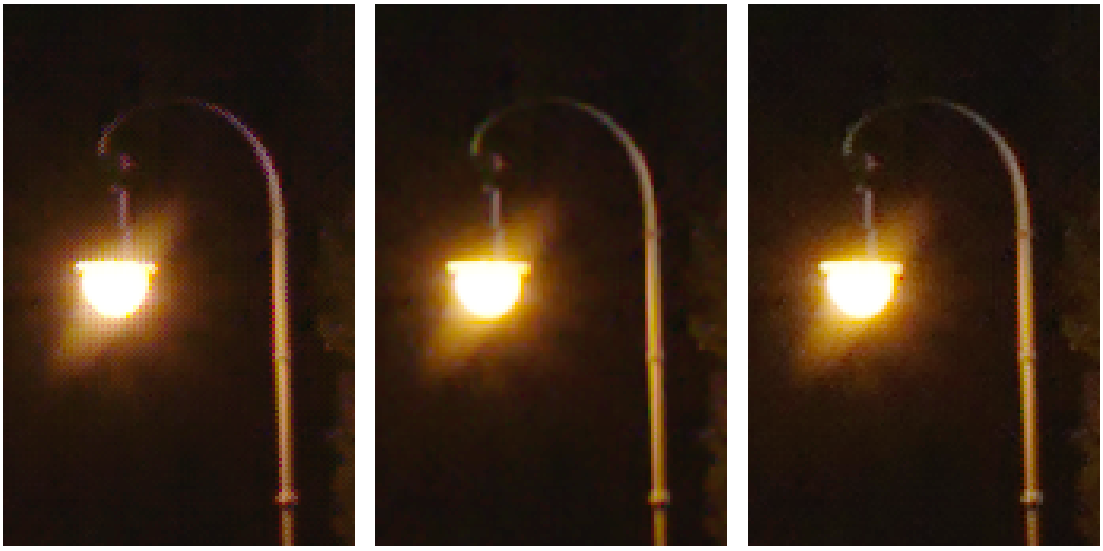
Method	JDD	JDD SD	RAD	Jeon	Zhang	LPA-ICI S/D Ada3	LPA-ICI CCF DLMMSE
PSNR	36.35	36.68	34.29	35.96	36.58	**38.51**	38.20

standard deviation and the LPA-ICI S/D method using a signal-dependent standard deviation that is obtained based on the measurements. Using the signal-dependence, more details are preserved in the dark parts of both buildings in the front.

Applying the LPA-ICI methods on real camera data leads to zipper artifacts at edges, and strong artifacts occur around highlights, as shown in Fig. 2.37. These artifacts can be avoided using more sophisticated debayering methods that are better optimized for camera data than the basic interpolation steps proposed in the original method. To reduce the artifacts around highlights for the simpler debayering methods, another solution can be used: a logarithmic transform can be applied before the LPA-ICI methods, which reduces the artifacts. However, this transform cannot eliminate the artifacts completely.

The PSNR of the city sequence was thereby increased up to 1.3 dB by using the camera debayering (Ada3) in the LPA-ICI S/D method (1 dB with DLMMSE) and up to 2 dB using the DLMMSE [130] in the LPA-ICI CCF method (1.5 dB with Ada3). The resulting PSNR values for the city sequence are compared to other methods from literature in Tab. 2.8. For comparison we used “Joint Denoising and Debayering” (JDD) by Menon et al. [83], “Regularization Approaches to Demosaicking” (RAD) by Menon et al. [84], Jeon entitles the method proposed by Jeon and Dubois in [58], Zhang the method by Zhang et al. in [131]. The methods are tested using the Matlab code from the authors. The algorithms usually assume a Gaussian distribution and a constant variance value. The JDD algorithm was adapted to the Alexa noise by taking into account the signal-dependent variances of the Alexa noise characteristic (JDD SD). For the other methods, we use the standard version with constant variance value. In order to obtain the best possible result the input parameters were iterated and the best obtained result was selected for comparison.

Both proposed LPA-ICI methods show similar visual quality, but the LPA-ICI CCF method has a significant lower cost in terms of required memory and arithmetic calculations in a hardware implementation than the LPA-ICI S/D methods. A good trade-off between the computational cost and the visual quality could be the modified LPA-ICI CCF method in combination with a performant debayer algorithm, as shown in Fig. 2.38.



(a) original **(b)** Ada3 **(c)** DLMMSE

Figure 2.38: LPA-ICI CCF is applied including the logarithmic transformation, and it is combined with different debayering methods: (a) original LPA-ICI CCF debayering, (b) including the camera debayering method Ada3, and (c) including DLMMSE debayering.

3 Denoising methods for real camera data

Building on the more realistic noise model from Chapter 2, two methods are presented in this chapter that include the more realistic camera noise model and can operate on linear data.

Both methods are computationally efficient, and could be implemented on a sufficiently powerful FPGA to achieve real-time processing. The first method builds on a DCT transformation. The DCT is quite commonly used in image processing on embedded systems. However, we have seen in the last chapter that a simple block DCT tends to produce artifact at edges and thus may not be the best choice. We therefore use a shape-adaptive transform (SA-DCT) in this work. Drawing from our previous work [3, 7], we propose in Sec. 3.1 to use the SA-DCT directly on Bayer data, thus avoiding the problem of spatially correlated noise due to debayering. We show how to include the model from Chapter 2 and include a low-cost extension for increasing the quality for video.

The second method builds on a wavelet transformation. Wavelet transformations lead to a sparse representation providing information about the frequency and locality of a signal. This can be very useful in denoising, as it allows retention of structure at edges while applying strong denoising in smooth image areas. Additionally, we identified an efficient possibility of including the spatial correlation of the noise after debayering. Section 3.2 first proposes a noise model in wavelet domain based on the analysis of the relevant processing steps from Chapter 2 and the steps additionally necessary for wavelet based denoising. The description builds on our previous publication [1]. We explore the possibilities with the simplest wavelet transformation: the discrete wavelet transformation (DWT) based on Haar filters. We build on the DWT despite the well-known fact that redundant transformations, e.g., the stationary wavelet transformation (SWT), provide a better signal representation enabling better denoising results. This is due to the implementation cost: the SWT would not be implementable with reasonable memory requirements, whereas the DWT may be suited for an efficient hardware implementation. We aim for a good trade-off between quality and implementation cost.

3.1 SA-DCT Denosing for Camera Raw Data

As mentioned in the introduction, denoising methods for camera data can be applied to the raw camera data, i.e., CFA data, or to the debayered full-color RGB data. Whereas in the latter case, the noise characteristics are more complex, the noise in the raw camera data has a known distribution and a signal-dependent variance, as explained in Chapter 2.2.1. The signal-dependent noise can be taken into account in the denoising method much easier than the complex and spatially correlated noise after debayering. Therefore, denoising methods for raw camera data are required.

However, despite the richness of denoising methods, not many methods have been proposed for raw data denoising up to now. Raw camera data denoising is challenging – due to the CFA, which is usually a Bayer pattern, neighboring pixels are not of the same color. Most denoising algorithms, however, rely on spatial redundancy, i.e., a local pixel neighborhood. As this neighborhood is not available in the raw data, existing denoising methods cannot be easily applied to raw camera data. CFA data denoising has been studied by Hirakawa et al., who extended a wavelet-based method to CFA data [51]. Zhang et al. proposed a principle component analysis (PCA) based solution [129]. The state-of-the-art method in image denoising, BM3D [33], was combined with a noise estimation algorithm and used for image denoising and declipping by Foi et al. [39]. The latter provides very good results, which, however, come at high computational cost. Nonlocal methods inherently have high memory cost, as the complete image must be available in the internal memory. With high resolution image sensors, this can be a limiting aspect.

This chapter therefore proposes a local method for raw data image and video denoising. The description builds on previous work: first results are discussed in [7], and a more detailed description with results using real camera data from the ARRI image set [11], as well as an improvement for video data, is presented in [3]. The method builds on a shape-adaptive DCT proposed by Foi et al. [40]. It relies on a homogeneous neighborhood for each pixel and a thresholding operation, which eliminates the noisy DCT coefficients. The neighborhood estimation prevents oversmoothing, which is the prevailing drawback of denoising. The same authors in [41] proposed an adaptation of the method for signal-dependent noise, thus it can be easily used for our noise model. However, the method is still not adapted to linear data and to CFA data. In the following, we propose a method to adapt this algorithm to the CFA data in the linear domain (Sec. 3.1.1). We calculate the neighborhood estimation on luminance data and we propose a luminance transformation that can be applied directly to the CFA data. Additionally, we show how to adapt the SA-DCT to Bayer data, as this is the most usual CFA, and describe how the real noise characteristics from a digital camera can be included in the method. Sec. 3.1.2 compares our solution both to Zhang et al. [129] and to Foi et al. [39] and evaluates the visual quality of image and video data. Sec. 3.1.3 presents the improvement for video data and discusses implementation aspects.

Table 3.1: PSNR results of different luminance estimation methods on the city image.

	Gaussian	h_L [57]	virtual	Ada3	LSLCD [57]
PSNR [dB]	36.84	36.74	36.76	36.85	36.66

3.1.1 Adaptive Raw data denoising

Our goal is to develop a denoising algorithm that provides high visual quality by remaining efficient in terms of hardware implementation costs. As proposed by Foi, to prevent both the edge blurring problems and the oversmoothing of the image, we use the LPA-ICI technique to find an adequate neighborhood for each pixel also considering that our algorithm must directly work on Bayer data, i.e., neighboring pixel do not belong to the same color channel. To estimate the neighborhood based on the Bayer data, we apply a luminance transformation. In color image denoising, a color space transformation from RGB to a luminance-chrominance color space (e.g., YCbCr) is commonly used. As the structural information in natural images is mostly contained in the luminance data, it is effective to perform the neighborhood estimation on the luminance channel only and use this neighborhood for denoising all three channels. In our case, we apply a similar strategy; we obtain an estimation of the luminance channel based on the Bayer data. We discuss this luminance transformation in the next section.

Luminance transformation of Bayer data

To estimate the luminance based on the Bayer data we tested different techniques: filtering with a fixed filter kernel, partial debayering, and a new method we call "virtual luminance". Partial debayering means, we directly use the debayered green channel as the estimated luminance, as the green channel is most dominant for the luminance. We used the camera debayering method (Ada3). Another low-cost luminance estimation is to apply filters directly to the Bayer data. We used two different filters: a Gaussian filter (3 × 3) and a filter similar to the luminance filter proposed by Jeon and Dubois in [57] (5 × 5). We additionally calculated a luminance directly on the Bayer data by using the neighboring color values, which we call "virtual" because the result provides luminance values which are located between the pixels.

The results on our test image "city" in Tab. 3.1 show that the difference in terms of PSNR of the denoising result is marginal. The best value is reached by the camera debayering and Gaussian filtering. We use the Gaussian filtering for our method, as it shows one of the best results and is a very simple and cost-efficient method.

LPA-ICI for neighborhood estimation

Once we obtained a continuous luminance estimation we can apply the LPA-ICI method to find the dimension of the local homogenous neighborhood. The LPA-ICI method chooses a polynomial model (LPA) of a certain scale. Based on the ICI rule, the scale of the model is chosen and this scale defines the extent of a shape around each pixel, in which no singularities or discontinuities are present.

The LPA-ICI method is applied in eight directions. In each direction θ_k a set of directional kernels g_{h,θ_k} with the varying scale h is used to find an interval D.

$$D_{\hat{y}_{h_i,\theta_k}} = \left[\hat{y}_{h_i,\theta_k} - \Gamma\sigma_{\hat{y}_{h_i,\theta_k}}, \hat{y}_{h_i,\theta_k} + \Gamma\sigma_{\hat{y}_{h_i,\theta_k}}\right] \tag{3.1}$$

$\Gamma > 0$ is a tuning parameter, which adjusts the size of the interval. The standard deviation of the estimate $\sigma_{\hat{y}_{h_i,\theta_k}}$ is calculated by multiplying the standard deviation σ of the input with the norm $\|g_{h,\theta_k}\|_2$ of the used kernel.

$$\sigma_{\hat{y}_{h_i,\theta_k}} = \sigma\|g_{h,\theta_k}\|_2 \tag{3.2}$$

The standard deviation of the input, σ, is calculated using Eq. 2.8 with the signal value x estimated by using the noisy observation, thus the raw data pixel value. This is a very simple estimate, but we found that the improvement using a better approximation is marginal.

The largest possible scale h_i is chosen using the ICI rule.

$$\mathcal{I}_{j,\theta_k} = \bigcap_{i=1}^{j} D_{\hat{y}_{h_i,\theta_k}} \tag{3.3}$$

This scale defines the shape dimension in the direction θ_k. The ICI rule is applied in all eight directions and thereby a neighborhood $\tilde{U}_x^+$ for the pixel position x is found. This neighborhood can now be used for denoising.

Shape-adaptive DCT and denoising via hard thresholding

The shape of the neighborhood is found for each pixel. Now the SA-DCT must be applied on the Bayer data to perform the denoising. Therefore the Bayer data is separated into the four sub-channels, R, G_1, G_2, and B, each of them containing one fourth of the total pixel number.

For each sub-channel the SA-DCT is implemented as proposed by Foi [40]. A local estimate $\hat{y}_{\tilde{U}_x^+}$ is obtained by thresholding in the SA-DCT domain with the threshold parameter t_x.

$$t_x = k_{thr}\sigma\sqrt{2\ln|\tilde{U}_x^+| + 1} \tag{3.4}$$

The constant k_{thr} regulates the denoising strength. The global estimate is given by a weighted average of the local estimates.

$$\hat{y} = \frac{\sum\limits_{x \in X} w_x \hat{y}_{\tilde{U_x^+}}}{\sum\limits_{x \in X} w_x \chi_{\tilde{U_x^+}}} \tag{3.5}$$

The weights w_x are calculated based on the size of the neighborhood $\tilde{U_x^+}$ and the noise variance σ^2. A smaller neighborhood gets higher weights, thus fine details are preserved. N_x^{har} is the number of non-zero coefficients after thresholding, thus sparse solutions are preferred over non-sparse solutions.

$$w_x = \frac{1}{\sigma^2 (1 + N_x^{har}) |\tilde{U_x^+}|} \tag{3.6}$$

We showed how to apply neighborhood estimation on Bayer data and discussed how this neighborhood can be used for denoising with SA-DCT hard thresholding. In the following we evaluate our method and compare it to other state-of-the-art methods.

3.1.2 Experiments: Image Quality versus System Performance

We compare our method to the state-of-the-art denoising method, BM3D [33], which was specifically adapted for raw data [39], and the PCA-based method from [129].

Although we also tested our method on real camera data, we compare our method quantitatively using simulated camera video sequences, as this provides us a realistic reference. This test method was described in [10] and we use it for our data similarly: we rendered the high-resolution image data and applied the camera simulation including the optical low-pass of the camera optics. We obtain a simulated reference image and including the camera noise added to the sensor data, we obtain realistic noisy images. These images are then denoised and compared to the reference. We tested the method using two different debayering methods: the camera debayering method Ada3 and the DDFAPD[1] by Menon et al. in [85].

We perform a test with different noise levels. In a digital camera the noise is signal-dependent corresponding to the characteristic described in Sec. 2.2.1. To obtain different noise levels, we change the simulated brightness of the image and subsequently process the images with a higher exposure index (EI) to obtain comparable results. This leads to a higher noise level, as the EI operates as a gain: the higher the EI, the higher the amplification and thus the lower the signal-to-noise-ratio. Three different noise levels were simulated: EI 800, which means a low noise level, EI 1600, and EI 3200, which corresponds to a quite high noise level.

Table 3.2: SA-DCT Bayer denoising results for the noise level EI 800.

EI 800	Debayering	PSNR	PSNR-HVS	SSIM	VIF
proposed	Ada3	40.390	36.766	0.993	0.777
proposed	DDFAPD	40.703	36.777	0.993	0.810
ClipFoi	Ada3	38.997	34.552	0.991	0.745
ClipFoi	DDFAPD	39.312	34.449	0.991	0.765
Zhang	Ada3	40.815	37.183	0.994	0.775
Zhang	DDFAPD	41.325	37.265	0.994	0.802

Table 3.3: SA-DCT Bayer denoising results for the noise level EI 1600.

EI 1600	Debayering	PSNR	PSNR-HVS	SSIM	VIF
proposed	Ada3	38.395	34.537	0.988	0.696
proposed	DDFAPD	38.682	34.787	0.988	0.737
ClipFoi	Ada3	40.649	37.058	0.995	0.792
ClipFoi	DDFAPD	41.233	36.794	0.994	0.818
Zhang	Ada3	38.957	34.821	0.990	0.676
Zhang	DDFAPD	39.458	35.089	0.990	0.703

Visual quality of denoising results

We first compare the method quantitatively based on quality metrics. We calculate the PSNR, as it is a very usual metric, and we use three additional objective quality metrics that according to [10] correlate better with the human perception of visual quality: a PSNR adapted to the human visual system (PSNR-HVS) [35], the structural similarity index (SSIM) [124] and the visual information fidelity (VIF) [112]. Tab. 3.2, 3.3, and 3.4 show the results for EI 800, EI 1600, and EI 3200, respectively.

The Bayer SA-DCT reaches the highest VIF for the EI 800 sequences. The results are very close to the PCA method by Zhang et al. [129], whereas the BM3D-based method by Foi et al. [39], called “ClipFoi” in the following, achieves lower results, which may be due to color artifacts. For higher EI, thus higher noise levels, the ClipFoi method reaches the highest metric results, but the Bayer SA-DCT and Zhang [129] show approximately the same results. We conclude that our method achieves competitive results with respect to usual quality metrics.

When comparing the results visually, however, we think our method provides slightly better results. As the visual perception is not well represented by the metrics, we can evaluate the results comparing the difference images

$$I_{diff} = I_{ref} - I_{denoised}.$$

[1] “Demosaicing With Directional Filtering and a posteriori Decision”

Table 3.4: SA-DCT Bayer denoising results for the noise level EI 3200.

EI 3200	Debayering	PSNR	PSNR-HVS	SSIM	VIF
proposed	Ada3	35.890	31.781	0.976	0.586
proposed	DDFAPD	36.120	32.239	0.979	0.631
ClipFoi	Ada3	39.272	35.458	0.991	0.726
ClipFoi	DDFAPD	39.690	35.237	0.991	0.752
Zhang	Ada3	36.342	31.678	0.980	0.546
Zhang	DDFAPD	36.726	32.031	0.981	0.567

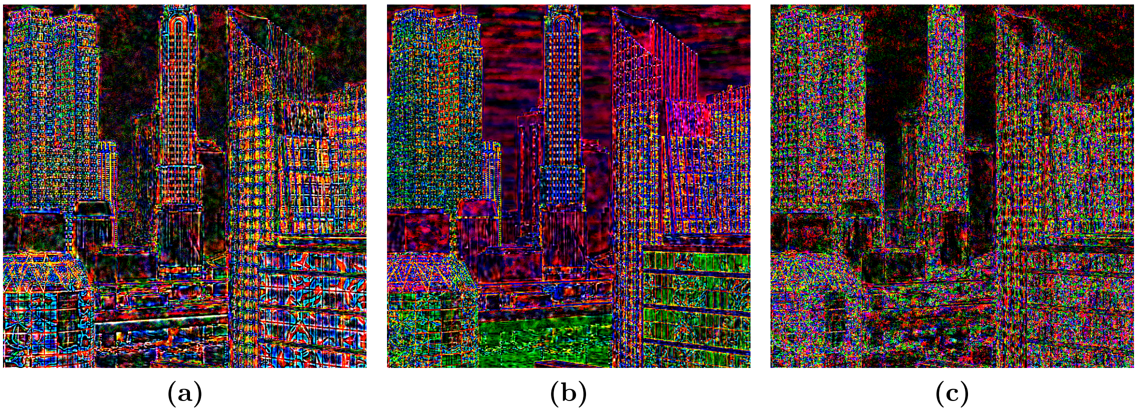
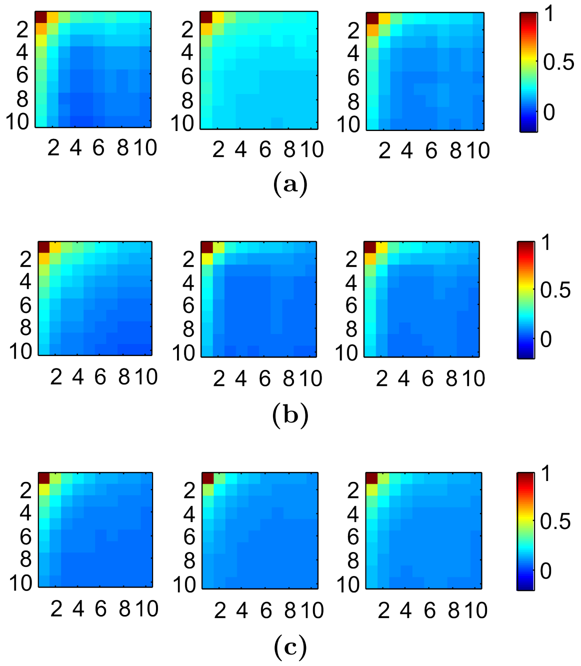


Figure 3.1: Difference to the reference for the test image “City” denoised using (a) the algorithm by Zhang [129], (b) Foi [39], and (c) the proposed version of the SA-DCT.

The difference images are also used to compare denoising algorithms, as suggested by Buades et al. [20]. An ideal difference image would look like noise; if image structure is visible, image details are lost during denoising. Fig. 3.1a shows the difference between the reference image and the image denoised by Zhang [129]: the image structure is clearly visible, which indicates that the image was blurred. Fig. 3.1b shows the difference image for the ClipFoi denoising result, in which we observe finer image structures, thus finer details seem to get lost. Additionally, the difference image is colored, which means that the error is an offset and thus the color of the denoised image does not correspond to the correct color of the reference. The difference image in Fig. 3.1c shows the difference for the proposed Bayer SA-DCT, which shows that the difference image is most similar to noise (it looks more random), hence, not much image structure is lost and the absence of offset indicates that also the colors are well preserved.

We calculated the correlation coefficient between the denoising error – also called “method noise” – at one pixel position and the denoising error at the neighboring pixel positions. Fig. 3.2c shows the results. The correlation was calculated for all



(a)

(b)

(c)

Figure 3.2: Spatial correlation of the difference images for red (left), green (middle) and blue (right) channel. Displayed is the correlation for a 10×10 neighborhood. From top to bottom: Foi [39], Zhang [129] and the proposed Bayer Data SA-DCT. Ideally only noise should be removed from the image and thus the correlation of the difference image should be low.

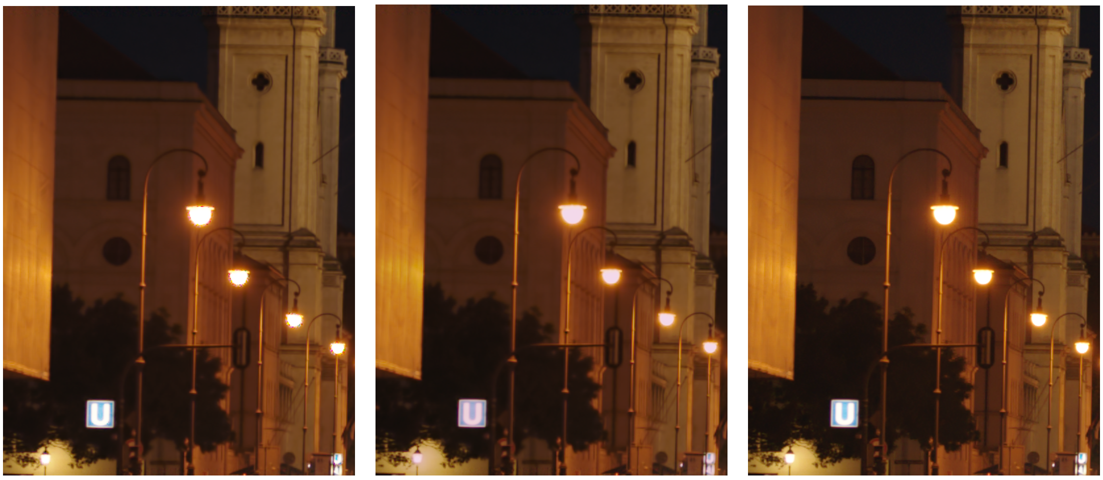
three color channels independently and shown for a 10×10 neighborhood. Furthermore, Tab. 3.5 provides the 3×3 correlation matrices of the three denoising methods (ClipFoi, Zhang and the proposed method) calculated based on the green channel. The numbers thus correspond to the correlation of the noise at one pixel position to the noise at the respective neighboring pixel position in the green channel.

Processing Real Camera Data

Applying image processing methods from literature to real camera data can lead to artifacts, which are due to the different characteristic of the real camera data compared to the standard test data. An investigation of the differences and resulting artifacts using debayering methods is provided by Andriani et.al. [11]. When we used the method by Zhang [129], we observed similar artifacts in the highlights, see Fig. 3.3a. To solve this, we first converted the data using a piecewise logarithmic transformation into the 12-bit logarithmic format, as described in [11]. Fig. 3.3b shows the result with the logarithmic transformation: the artifacts around the highlights are almost not visible anymore. However, there are strong color shifts as it can be seen in the ’U’ sign and in the bottom right corner in Fig. 3.3b. The result

Table 3.5: Spatial correlation of the difference images in the green channel calculated in a 3x3 neighborhood.

$$C_{Foi} = \begin{bmatrix} 1 & 0.5672 & 0.4167 \\ 0.5893 & 0.4047 & 0.2999 \\ 0.4104 & 0.2807 & 0.2645 \end{bmatrix} \quad C_{Zhang} = \begin{bmatrix} 1 & 0.4795 & 0.2767 \\ 0.5235 & 0.2826 & 0.1571 \\ 0.3119 & 0.1545 & 0.0935 \end{bmatrix} \quad C_{SA-DCT} = \begin{bmatrix} 1 & 0.3731 & 0.2359 \\ 0.4277 & 0.2449 & 0.1846 \\ 0.2745 & 0.1923 & 0.1534 \end{bmatrix}$$


(a) PSNR 35.7 dB

(b) PSNR 37 dB

(c) PSNR 41.1 dB

Figure 3.3: Denoising results for the test image night by using the algorithm of Zhang [129] in linear domain (a), in logarithmic domain (b), and by using the proposed version of the SA-DCT in (c).

of the proposed method is shown in Fig. 3.3c. We additionally calculated the PSNR of the real camera sequences using a reference image calculated as the mean image over 240 frames of a static sequence. The PSNR is clearly higher, as the proposed method does not show the mentioned color artifacts and effectively removes the noise in the image.

3.1.3 Video Sequences

When comparing video sequences we observe that both Foi's and Zhang's method show more temporal flickering than the proposed Bayer SA-DCT. This is probably due to the lower spatial correlation of the remaining noise, shown in Tab. 3.2c. Low spatial correlation means fine-grain noise, which is less visible than coarser grain noise due to the frequency-dependent color sensitivity of the human visual system. However, while the correlation is lower in the proposed SA-DCT compared

Table 3.6: SA-DCT Bayer denoising results without temporal denoising. The metrics are calculated and averaged over 10 frames.

Sequence	EI	PSNR	PSNR-HVS	VIF	MSSIM
Color Test Chart	800	43.72	44.31	0.4912	0.9963
Night Odeonsplatz	800	41.18	39.33	0.7175	0.9949
Siegestor Night	1600	39.89	39.22	0.6280	0.9911

Table 3.7: SA-DCT Bayer denoising results with the temporal denoising step. The metrics are calculated and averaged over 10 frames.

Sequence	EI	PSNR	PSNR-HVS	VIF	MSSIM
Color Test Chart	800	44.63	44.88	0.5008	0.9976
Night Odeonsplatz	800	41.77	39.41	0.7308	0.9958
Siegestor Night	1600	40.15	38.87	0.6492	0.9924

to the other two methods, we still observe some temporal flickering in our results. Therefore we extend our method using an additional temporal denoising step.

The flickering could be a consequence of the mean calculation, which dominates the result for large homogeneous areas during the denoising. Large homogeneous areas are likely to have maximum size $|\tilde{U}_x^+|$, which results in a large threshold and hence the result is mostly dominated by the mean and the DC-coefficient. As the DC-coefficient is not noise-free, temporal flickering can be observed.

To improve the visual quality in video sequences, we therefore propose to add an additional temporal denoising step. As motion estimation is difficult on Bayer data, we propose to use a very simple criterion for motion detection: if the deviation from a pixel in one frame to the same pixel in the next frame is large, we do not use it because we assume that the pixel changed due to motion. If the pixel is similar, we use it for the temporal denoising step. A reasonable threshold depends on the noise variance.

To reduce the flickering, we average over n past and future frames. The temporal smoothed output $S_{x,i}$ for every pixel x of the current frame $I_{x,f}$ can be calculated based on the past and future frames $I_{x,i}$ as follows:

$$S_{x,f} = \frac{\sum\limits_{i=f-n}^{f+n} I_{x,i}\Phi(I_{x,i}, I_{x,f})}{\sum\limits_{i=f-n}^{f+n} \Phi(I_{x,i}, I_{x,f})} \tag{3.7}$$

$$\Phi(I_{x,i}, I_{x,f}) = \begin{cases} 1 & \text{if } I_{x,i} - I_{x,f} \leq \tau(I_{x,f}) \\ 0 & \text{else} \end{cases} \tag{3.8}$$

The threshold τ linearly depends on the noise standard deviation σ, which is estimated based on the realistic model described in Sec. 2.2.1.

$$\tau(I_{x,f}) = k \cdot \sigma(I_{x,f}) \tag{3.9}$$

Tab. 3.6 shows the result of the single-image SA-DCT without temporal denoising and Tab. 3.7 shows the results of the SA-DCT denoising with additional temporal denoising. The temporal denoising step leads to a higher PSNR and effectively reduces the flickering of the video sequence. The visual quality of the video sequence is therefore clearly higher with the temporal denoising step.

Implementation aspects

Denoising algorithms like BM3D usually assume that the complete noisy image is available at each pixel position. In stream-based image processing, as it is usual in embedded systems and communication, only one pixel at a time is available. In these applications an additional buffer must be implemented to provide the neighborhood for the denoising step. Therefore, the memory cost is quite high for algorithms like BM3D. Our method requires only a local neighborhood for the denoising step and is therefore better suited for stream-based processing. Additionally, it operates on Bayer data, and the raw Bayer data has only one value per pixel whereas the processed RGB data has three values per pixel. Thus, a three times lower complexity can be expected compared to algorithms that require the fully processed image data. The temporal denoising step implementation requires additional frame buffers, but the computational cost of the calculation is kept very low.

3.1.4 Conclusion

We proposed a method for real camera Bayer data denoising based on a neighborhood estimation combined with a shape-adaptive DCT. Although the method has been proposed for RGB or grayscale image data, it couldn't be applied to Bayer data directly. To perform the SA-DCT on Bayer data, we apply LPA-ICI-based neighborhood estimation to the luminance data. As the luminance data is not available in the Bayer data, we estimate the luminance efficiently using different methods. The best tradeoff between computational cost and quality was found using Gaussian filtering. Based on the neighborhood estimation, a hard thresholding is performed on the coefficients of the shape-adaptive DCT. The threshold includes the noise variance, and we show how real camera noise characteristics can be integrated. To evaluate our method, we compare it with two state-of-the-art algorithms: a PCA-based CFA denoising and a BM3D-based denoising that uses noise variance estimation. Our method achieves competitive results in terms of PSNR and can lead to better visual quality with lower computational cost. An additional temporal denoising step is proposed, which effectively reduces temporal flickering in real camera video sequences.

3.2 Wavelet Denoising of linear debayered data

As mentioned in the introduction, two main issues need to be addressed for effective noise reduction that can be applied to nowadays real camera data. First, as the pixel count of today's sensors is increasing, the data rate and memory requirements are increasing significantly. Thus, efficient methods need to be developed that can handle video data with a resolution of HD, 4k and beyond. Second, the usual assumption that noise can be modeled as additive white Gaussian noise (AWGN) does not hold in the case of real camera noise.

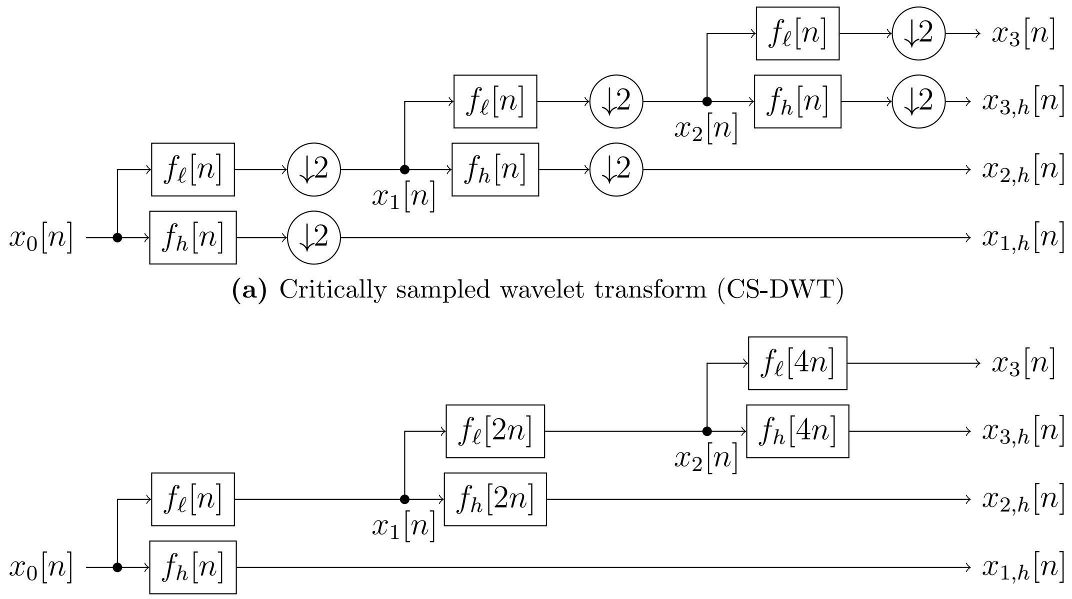
To solve both tasks, wavelet based denoising methods seem promising. First, the scalable properties of the wavelet transformation allow accounting for available resources. To achieve a low-cost denoising method we use the extremely low cost wavelet transformation, the classical critically sampled discrete wavelet transform (DWT), and the simplest filters, the Haar filters. Second, a wavelet based representation facilitates to account for real camera noise, as the signal is decomposed into several scales that represent different frequency bands, and hence non-white noise can be respected using level-dependent operations.

The implementation of wavelets for image processing has been discussed and advanced; in Hsia et.al. [53] an overview can be found. Although undecimated wavelet transforms are better for denoising, the memory cost of these transforms is extremely high and increases with the image size. Although some implementations of undecimated wavelet transforms have been proposed, e.g., [62], the memory cost of these transforms can be expected to be too high for the high resolution video data of today's digital cameras. This section hence focuses on the low cost decimated transforms. The computational cost of the denoising operation should also be low; hard thresholding and bivariate thresholding is used, the latter uses the inter-scale dependency of the wavelet coefficients.

Based on our publication [1], this section starts with providing a short overview of the wavelet decomposition methods and the denoising methods. Sec. 3.2.2 explains the camera processing and how to obtain data describing the noise of the different color channels and the different frequency bands. Furthermore, it shows how this data can be used in two wavelet denoising methods: simple hard thresholding and bivariate thresholding [111]. Results are presented in Sec. 3.2.3.

3.2.1 Wavelet Denoising

Wavelet denoising can be separated into three steps: the wavelet decomposition, denoising the wavelet coefficients and the inverse wavelet decomposition. The different types of wavelet transforms are discussed and subsequently we provide an overview of possible solutions for the denoising step.



(a) Critically sampled wavelet transform (CS-DWT)

(b) Stationary wavelet transform (SWT)

Figure 3.4: Structures of different wavelet filter structures

Wavelet Decomposition

Many other decomposition methods, such as the discrete cosine transform, have a very specific definition and a limited number of parameters. The term wavelet decomposition, however, does not correspond to one single filter. Instead, multiple different decompositions have been proposed, differing in their intended application and computational cost.

Often, the term DWT (discrete wavelet transform) is used to refer to a particular filter structure, shown in Figure 3.4a. The input signal $x_0[n]$ is filtered by two different filters, $f_\ell[n]$ and $f_h[n]$. The results are downsampled by a factor of two, yielding the signals $x_1[n]$ and $x_{1,h}[n]$. By design, $x_1[x]$ contains the low-pass components of $x[n]$ and $x_{1,h}[n]$ the remaining high-pass components. The procedure is repeated on $x_1[x]$ in order to build a hierarchical filter. After all N levels, the final low-pass coefficients $x_N[n]$ are obtained, called the approximation coefficients.

The filters are chosen such that perfect reconstruction of x_i from x_{i+1} and $x_{i+1,h}$ is possible for all levels i. Therefore, it is sufficient to keep only the high-pass coefficients $x_{1,h}, x_{2,h}, \ldots$ in addition to the approximation coefficients for perfect reconstruction of the input signal. Due to the downsampling steps, the number of coefficients is the same as the number of samples in the input signal. Therefore, we refer to this kind of wavelet transform as critically sampled discrete wavelet transform (CS-DWT), or shorter DWT, in the remainder of this dissertation.

$f_h[n]$ and $f_\ell[n]$ are usually finite-impulse filters. The simplest non-trivial filters that satisfy the perfect reconstruction property are known as the Haar filters. Alternatively, the Daubechies construction method can be used to produce filter kernels of any length, which have stronger high-pass and low-pass characteristics than the Haar transform, but are more costly to implement.

As a consequence of the downsampling steps, a shifted signal $x[n-1]$ does not usually yield shifted coefficients. When a lossy operation is performed on the coefficients, this can manifest itself as blocking artifacts. A simple approach to solve this problem is to omit the downsampling steps entirely. The result is called the stationary wavelet transform (SWT), illustrated in Figure 3.4b. While the CS-DWT uses the same filters at every level, the SWT upsamples the filters at every step to compensate for the missing downsampling steps.

To achieve a low-cost denoising method implementable in real-time applications, we use the most simple wavelet structure, the CS-DWT, and the simplest filters, the Haar filters.

Denoising

Thresholding A basic approach for denoising wavelet coefficients is a coefficient-wise thresholding of all high-pass signals. The simplest thresholding method is known as hard thresholding.

$$w_{\text{hard}}(x) = \begin{cases} 0 & \text{if } |x| < \tau, \\ x & \text{if } |x| \geq \tau, \end{cases} \tag{3.10}$$

$\tau \geq 0$ is the threshold value; $w_{\text{hard}}(x)$ is the resulting wavelet coefficient. The wavelet coefficients of an image are assumed to be sparse, thus most of the signal is contained in relatively few coefficients with a high absolute value, whereas the noise is spread uniformly across all coefficients.

Higher values of τ cause stronger denosing, but also incur higher losses of signal data. The optimal value depends on the underlying wavelet transform, the amount of noise in the input signal and the metric used to determine optimality. Thresholding methods are easy to implement using basic adders and comparators. To account for the camera noise variance, the fixed threshold in the standard method is replaced by the threshold $\tau = k\sigma$. Thresholding methods are easy to implement using basic adders and comparators.

Bivariate Denoising As large coefficients tend to propagate across the decomposition scale, a significant wavelet coefficient c_1 is usually related to a high parent coefficient c_2, which is the coefficient at the same location in the lower frequency band. This means, if the parent coefficient is large, the child contains generally more details and thus should be denoised with caution. To find an estimate r_1 for the coefficient c_1, a shrinkage function is applied based on the parent (c_2) and the child (c_1) wavelet coefficient in the bivariate denoising method proposed by Sendur et.al. [111].

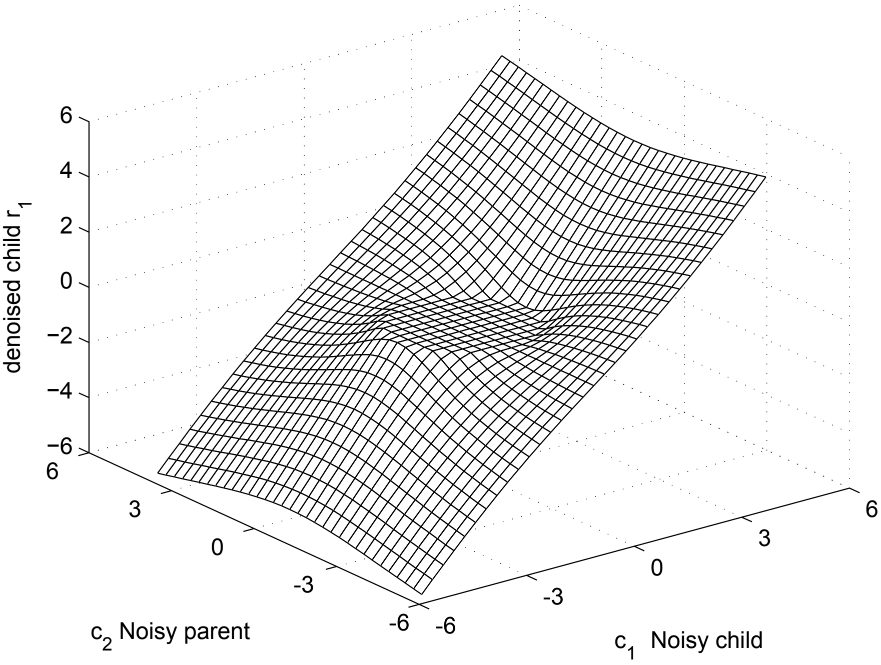


Figure 3.5: Bivariate shrinkage function proposed in equation (3.11).

$$r_1 = \frac{\left(\sqrt{c_1^2 + c_2^2} - T\right)_+}{\sqrt{c_1^2 + c_2^2}} \cdot c_1. \tag{3.11}$$

Fig. 3.5 illustrates this shrinkage function. The threshold T additionally takes into account the noise variance σ_n^2 and the signal variance σ_s, which is calculated from the equation: $\sigma_{c_1}^2 = \sigma_s^2 + \sigma_n^2$.

$$\sigma_s = \sqrt{\left(\sigma_{c_1}^2 - \sigma_n^2\right)_+}. \tag{3.12}$$

$\sigma_{c_1}^2$ is calculated based on its neighborhood $N(c_1)$ of size M.

$$\sigma_{c_1}^2 = \frac{1}{M} \sum_{c_j \in N(c_1)} c_j^2. \tag{3.13}$$

The original bivariate method [111] uses the following threshold T_{orig}.

$$T_{orig} = \frac{\sqrt{3}\sigma_n^2}{\sqrt{\left(\sigma_{c_1}^2 - \sigma_n^2\right)_+}} \tag{3.14}$$

The threshold T_{orig} becomes very large when the noise variance σ_n is close to the variance $\sigma_{c_1}^2$, which is estimated based on the neighborhood. This yields that the signal variance σ_s decreases towards zero. Hence, as the denominator is almost zero, the threshold T_{orig} becomes extremely high, and even very high coefficients are set to zero. This is especially critical at edges, where this effect causes strong artifacts in the denoised camera data.

To avoid this problem we propose an adapted threshold function.

$$T = \frac{\sqrt{3}\sigma_n^2}{\sigma_s + \sigma_n} \tag{3.15}$$

In comparison to the original threshold, we added σ_n to the denominator. The new threshold is thereby limited by an upper bound of $\sqrt{3}\sigma_n$. This ensures that large signal coefficients are not deleted. Thereby we retain more details in the images and obtain fewer artifacts around edges.

In the original method, σ_n^2 is estimated via the standard median estimator calculated on the first scale diagonal coefficients c_{D1}: $\sigma_n^2 = \frac{\text{median}\ (|c_{D1}|)}{0.6745}$. To adapt the method to real camera noise, we replace the median estimator by a more realistic noise variance, which is based on analyzing the camera processing described in Section 3.2.2.

Implementation

In the above described approach, the main resources are consumed by the wavelet decomposition. Thus, we focus on the decomposition implementation in this section. Many hardware implementations of the discrete wavelet transform have already been published. However, none of these are directly suited to the way data is transported in stream-based video processing.

One-dimensional wavelet transform units consume relatively little resources [60, 61]. In the simplest case, one can directly implement the filters in Figure 3.4. Due to the downsampling steps, the higher-level filters run at lower frequency than the ones for the first-level transformation. If all filters are implemented by identical hardware modules, the available processing capabilities are thus underutilized. To improve efficiency, all filters can be implemented in one single hardware module with input multiplexers, which causes that module to be almost fully utilized [119].

Transitioning to two-dimensional filtering means that the filters do not only have to be applied along the rows of the image, but also along the columns. Many implementations assume the availability of a memory that can store the entire image or at least sufficiently-sized blocks, where any pixel can be read on demand [103]. In this way, the filters can always be fed the appropriate row or column of the image. However, this is not possible with nowadays high resolution video data.

A single block of an image will easily fit into the internal memories of a modern FPGA, but in stream-based image processing data is received as a row-wise pixel stream. We therefore cannot select a single block to process, but instead need to store enough blocks to fill the whole width of the image. Additionally, roughly the same amount of memory has to be set aside for buffering in order to keep up the continuous intake and output of image data.

A prominent usage of wavelets is related to compression algorithms such as those specified in the JPEG2000 standard. There are hardware implementations tailored to this usage [34]. These are, however, not real-time capable and generally require the assistance of a CPU to provide the data in the correct order.

Although the lifting scheme [72, 93] provides a way to create more efficient implementations for longer filter kernels, the critical resource in stream-based image processing is not the computation resources but rather the memory usage, which cannot be addressed by lifting.

The completed design of CS-DWT was synthesized and tested on an evaluation board and a timing analysis showed that the unit is capable of running at clock speeds exceeding 200 MHz. The critical path lies in the control state machines, which could probably be optimized at the cost of increased logic utilization. The achieved speed is easily sufficient for real-time video processing.

Synthesis was run using different sets of parameters. As a baseline, a 3-level Haar Transform (filter length two) has been implemented for three-channel images with 2048 pixels per row and 18 bits per channel. Additionally, a version with two wavelet levels has been evaluated. Whereas logic and DSP resources are sufficiently available in today's FPGAs, the RAM usage increases exponentially with the number of levels. While the two-level decomposition can easily be implemented with the typically available RAM blocks, in the three-level decomposition the required memory resources are very high. As about half of the RAM is used by the delay unit, whose size increases exponentially with the number of levels, higher levels have not been tested. An external memory would be needed to realize higher level wavelet decompositions. Since an external memory chip can provide more memory than available internally by orders of magnitude, and the resource usage of the remaining units hardly depends on the number of levels, five levels of wavelet transform could easily be achieved.

3.2.2 Camera noise

To adapt the presented denoising operations from Section 3.2.1 to real camera noise, we need to analyze the noise characteristics in detail. Although the noise in the sensor data of a digital camera is Poisson noise [121], it can be well approximated by a Gaussian distribution with a signal-dependent variance [5]. However, the sensor data cannot be used directly for denoising in wavelet domain. It must be white balanced and demosaicked to obtain an RGB image (details can be found in the Sec. 2.1). After these steps the simple signal-dependent noise model cannot be used due to the spatial and chromatic correlation. Building on Sec. 2.2.3, which presents an analysis that showed the significant impact of debayering on the noise of the wavelet coefficients, a detailed analysis for all the steps previous to denoising is shown in this section.

The processing pipeline including a wavelet-based denoising step is shown in Fig. 3.6. As additional processing steps lead to a more complex noise characteristic (see Sec. 2.2.4), we apply the denoising step as soon as possible in the processing pipeline. Although denoising methods exist that operate on Bayer data [51], the application of wavelet denoising on real camera Bayer data can lead to artifacts. Especially color artifacts on edges can appear. The usual input data for wavelet-based denoising is a decorrelated Luma-Chroma color space, e.g., the simple and often used YCbCr color space. Thus, the necessary steps that are applied before denoising are the white balance, the debayering, a color space conversion, and the wavelet decomposition.

Analysis of the camera noise in the processing pipeline

To improve wavelet denoising methods without significant computational cost, we propose to include only the variance of the camera noise into the denoising methods and disregard the exact shape of the distribution. In the following, we explain the variance changes through the necessary processing steps applied before denoising. Subsequently, we show how to integrate this information into the wavelet denoising methods in Section 3.2.2.

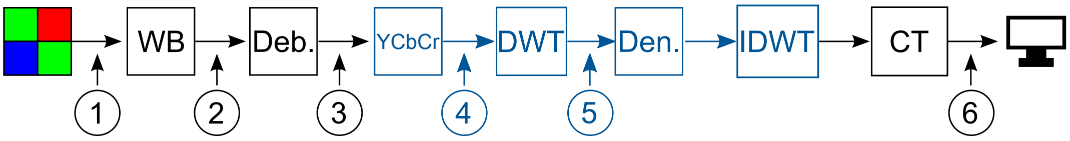
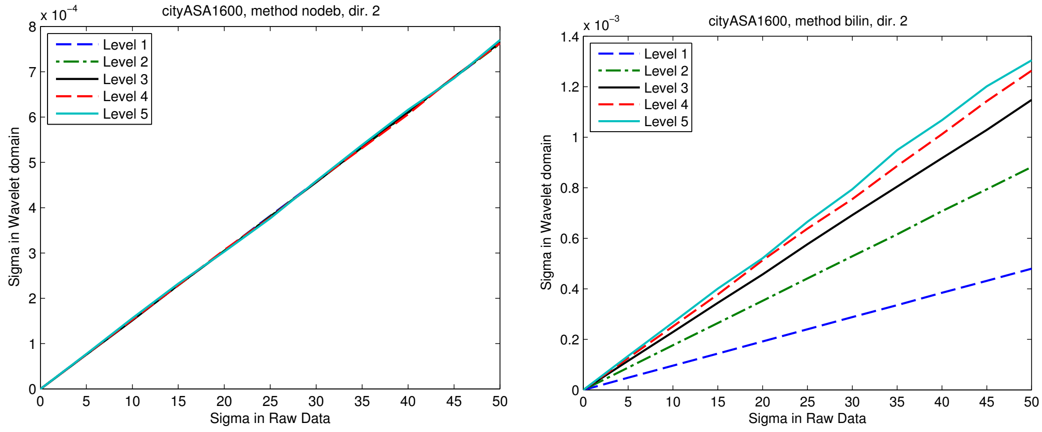


Figure 3.6: The processing steps including the wavelet denoising step are the white balance (WB), debayering (Deb.), the YCbCr transformation, the discrete wavelet transform (DWT), the denoising step (Den) and the remaining color transformations (CT).



(a) no deb.

(b) bilinear

Figure 3.7: The noise variance of the wavelet coefficients over the input variance is shown for with (right) and without debayering (left) applied before the decomposition. The noise in the high frequency bands (Level 1 and 2) is diminished by debayering, while the noise in the higher levels is slightly stronger.

White Balance and debayering The first step, the white balance, is a linear gain factor g_c on each of the R-, G-, and B-channels. The debayering is the second step. It generates three color values by interpolation using the pixel value and the neighboring pixel's values. Therefore, a spatial and chromatic correlation in the resulting three color channels is introduced. The spatial correlation of the noise n_{deb} in the debayered image is usually disregarded in common state-of-the-art denoising methods [5]. While it may be difficult to include the spatial correlation in pixel domain denoising methods, wavelet denoising methods are a good choice, because the spatial correlation leads to a different amount of noise in the different frequency bands, like already shown in Sec. 2.2.3. As one frequency band corresponds to one wavelet level, we use level-dependent variance information to incorporate the spatial correlation of the noise after debayering.

YCbCr transformation After debayering, the subsequent step is the YCbCr transformation. The variance after a matrix transformation is well described in [22]. The matrix transformation is a linear combination of the color values, and thus, we can use linear noise gain factors to integrate the variance changes through the YCbCr transformation. However, the calculation must build on a correct correlated noise model to provide correct results.

Wavelet decomposition On the YCbCr data the wavelet decomposition is applied. This is the last step to obtain the wavelet coefficients, which are the input data for our denoising step. The wavelet transformation is applied using a filter bank, as

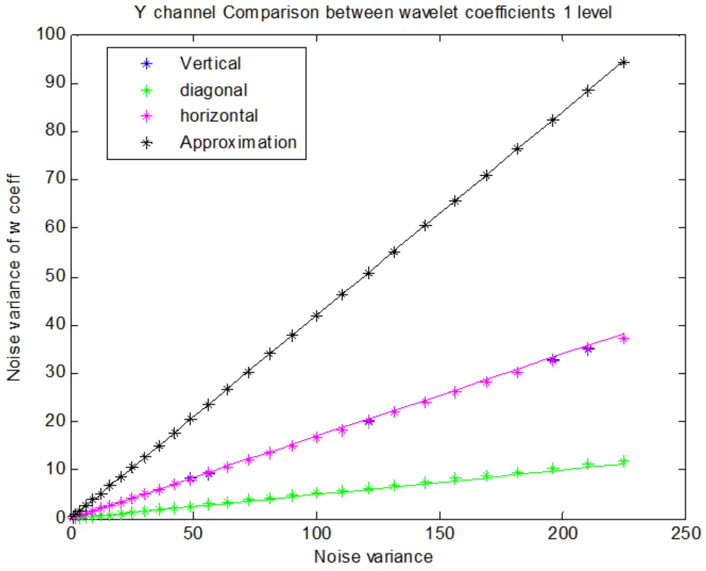


Figure 3.8: Approximation of the output variance after debayering and DWT in the Y channel.

described in Section 3.2.1. With the most simple filter type, the Haar filter, the filtering is a mean and a difference signal. For an uncorrelated input, the variance of the filter outputs can be easily calculated. However, the signal is spatially correlated due to debayering before the transformation.

To show the effect of debayering on the noise variance in the wavelet levels, we applied artificial noise to a computer-generated image and plotted the results of the different wavelet bands in Fig. 3.7. In a second test we applied debayering to the same noisy image. After debayering, the noise in the high frequency bands (Level 1 and 2) is diminished by debayering, because of the interpolation step. The noise in the higher levels is slightly stronger. As the effect is quite linear, we can account for it using a level-dependent noise gain factor.

Noise gain factors

The calculation of this variance factors can be complicated as the theoretical investigation becomes more complicated and more error prone when more steps are concatenated and a complex debayering algorithm is used. We thus choose an easier way to obtain the noise variance after all the steps: to evaluate the complete variance changes of all these four steps realistically, we perform a simulation and measure the variance of the output. This method can easily be repeated with different debayering algorithms, whereas the theoretic calculations are different for each algorithm and can be very time-consuming.

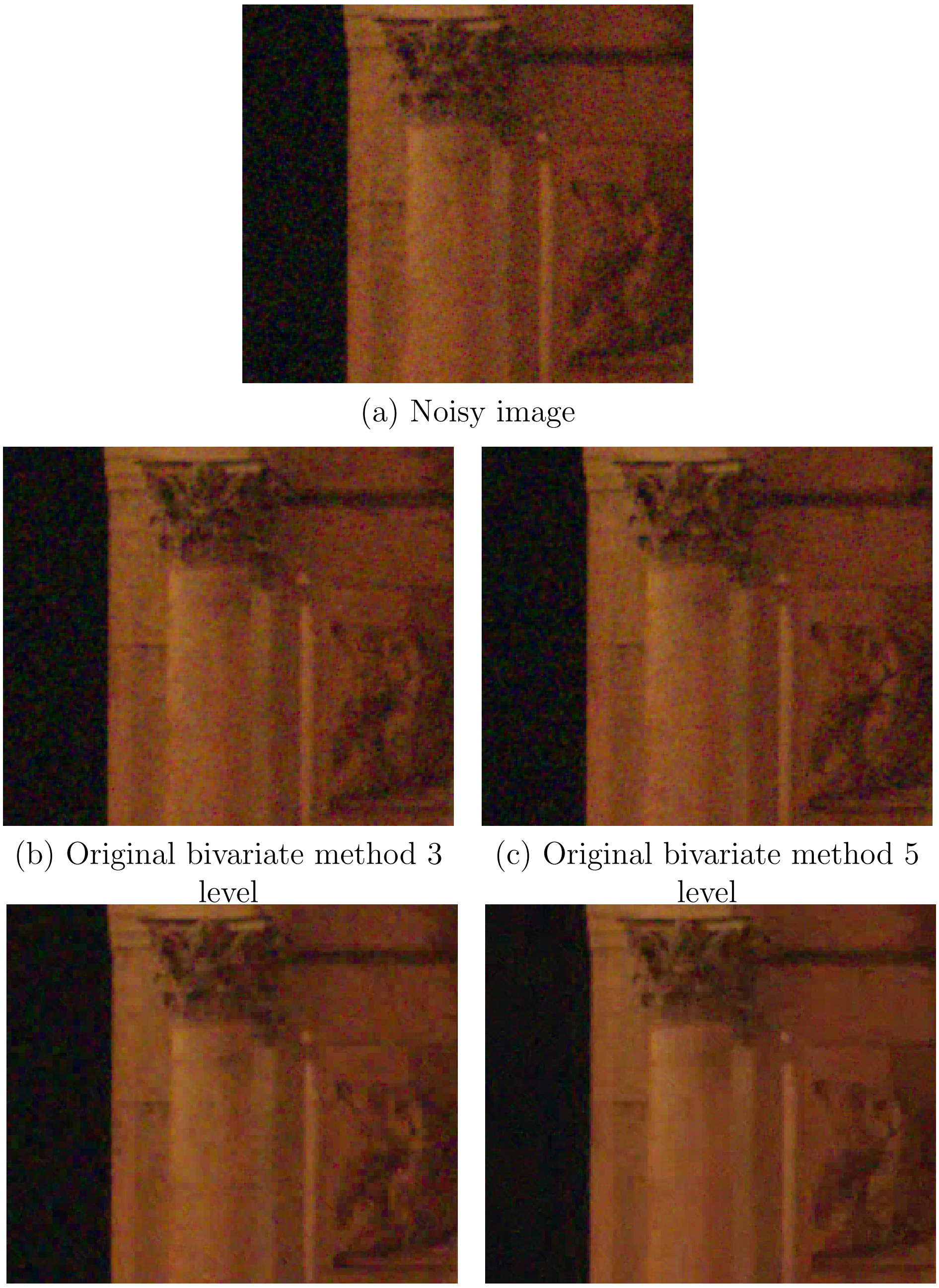
In this simulation we apply all the steps before the denoising operation, as shown in Fig. 3.6, to obtain the variance of the denoising input signals, thus the variance of the wavelet coefficients in Y, Cb, and Cr channel. We apply artificial noise with a variance σ_n^2 to homogeneous image data at sensor level, and investigate the variance of the wavelet coefficients $\sigma_{w,n}^2$ for each color channel (Y,Cb,Cr), each wavelet direction (vertical, horizontal, diagonal), and each wavelet level. Fig. 3.8 shows the input vs. coefficient variances for the first wavelet level in the Y channel. As the curves are quite linear a noise gain factor is sufficient to model the noise in the wavelet coefficients. Tab. 3.8 shows the noise gain factors calculated based on the simulation results. The noise variance $\sigma_{w,n}^2(l, d, c)$ in the wavelet coefficients is direction sub-band (d), level (l), and channel (c) dependent. Using these noise gain factors we can account for real camera noise in the wavelet denoising methods.

Table 3.8: Noise gain factors for the Debayering step, YCbCr and the wavelet transformation. σ_n^2 is noise variance before Debayering step.

Y channel	Approximation	horizontal-vertical	diagonal
1st level dec.	0.42 σ_n^2	0.17 σ_n^2	0.05 σ_n^2
2nd level dec.	0.62 σ_n^2	0.40 σ_n^2	0.27 σ_n^2
3rd level dec.	0.7 σ_n^2	0.59 σ_n^2	0.51 σ_n^2
4th level dec.	0.75 σ_n^2	0.70 σ_n^2	0.66 σ_n^2
Cb channel	Approximation	horizontal-vertical	diagonal
1st level dec.	0.44 σ_n^2	0.12 σ_n^2	0.03 σ_n^2
2nd level dec.	0.71 σ_n^2	0.41 σ_n^2	0.24 σ_n^2
3rd level dec.	0.84 σ_n^2	0.7 σ_n^2	0.56 σ_n^2
4th level dec.	0.91 σ_n^2	0.84 σ_n^2	0.77 σ_n^2
Cr channel	Approximation	horizontal-vertical	diagonal
1st level dec.	0.47 σ_n^2	0.14 σ_n	0.04 σ_n^2
2nd level dec.	0.74 σ_n^2	0.40 σ_n	0.3 σ_n^2
3rd level dec.	0.9 σ_n^2	0.7 σ_n	0.6 σ_n^2
4th level dec.	1 σ_n^2	0.9 σ_n	0.8 σ_n^2

3.2.3 Experimental Results

We first evaluate the objective quality using known image quality metrics. To reliably evaluate our method, we need real camera data. However, we also need the corresponding reference data to compare with and calculate the objective quality metrics. We use five different camera sequences with static image content, such that we can obtain a reference image by averaging. To obtain test data, we apply noise to the linear Bayer data and process the data according to Figure 3.6, thus apply white balance and debayering. To evaluate our method for channel-independent noise, we apply only the debayering step without the white balance. For both tests (with and



(a) Noisy image

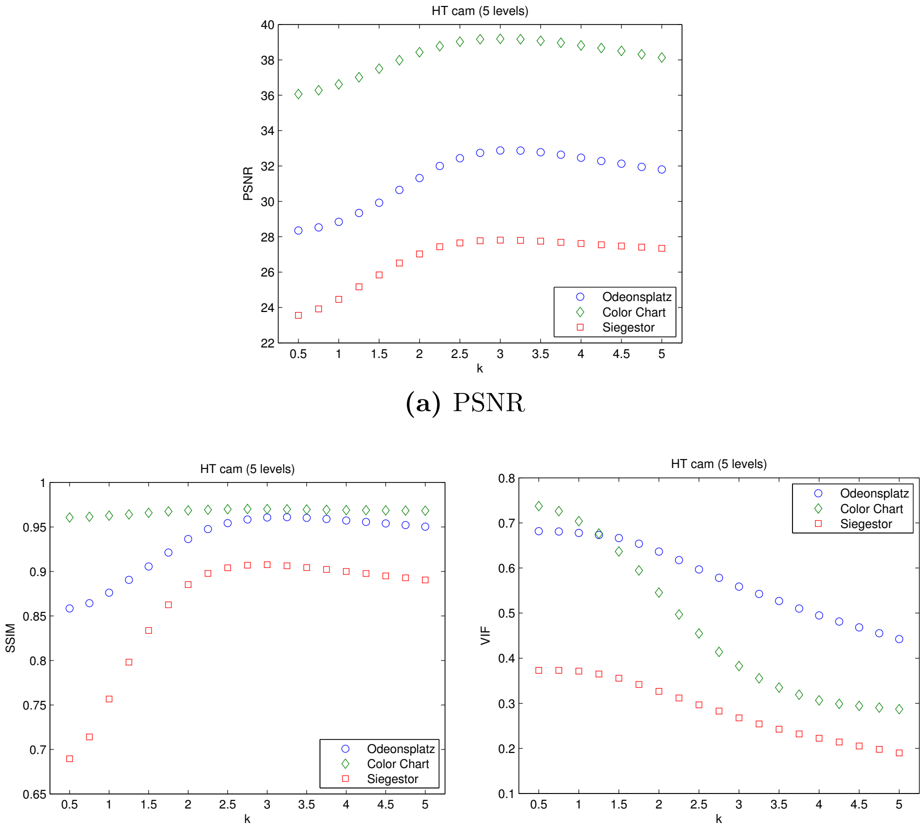
(b) Original bivariate method 3 level

(c) Original bivariate method 5 level

(d) Proposed method 3 level

(e) Proposed method 5 level

Figure 3.9: The noisy image and the denoised images obtained by applying denoising after three and five decomposition levels with the original bivariate method and the proposed method.



(a) PSNR

(b) SSIM

(c) VIF

Figure 3.10: Metric results for wavelet hard thresholding with a threshold $\tau = k\sigma$.

without white balance) the set of noise gain factors is calculated as described in the previous section.

Whereas the bivariate method does not need a parameter to be set, the hard thresholding must be optimized. Therefore, we perform a parameter sweep for each image to find the optimal denoising result. The optimal parameter thus is different for each metric. The PSNR is still the most widely used metric in image denoising, however, as the PSNR does not correlate well with the human perception of image quality, other metrics, e.g., the structural similarity index (SSIM) [124] and the visual information fidelity (VIF) [112] have been proposed. Fig. 3.10 shows the results of the parameter sweep with the adapted wavelet hard thresholding. The fixed threshold in the standard method is replaced by the threshold $\tau = k\sigma$.

We observe that the PSNR maximum is mostly around $k = 2$ up to $k = 3$. As most of the noisy values corresponding to the same mean value will be in the interval of 2σ (68%) or 3σ (95%) this means that most of the noise is eliminated. When we look at the denoising results, we observe, however, that most details are eliminated, and strong artifacts occur in the PSNR-optimized denoising result. To

Table 3.9: PSNR in [dB] for the different algorithms with equal noise in R, G and B channel, thus without the white balance step, $\sigma_n = 10$.

Sequence	Std. HT	Camera HT	Std. Bivariate	Camera Bivariate
Siegestor	30.93	31.03	26.13	30.01
Odeonsplatz	34.06	34.35	32.18	34.93
Ball Pool	33.54	33.19	31.19	34.02
Tree & Lights	27.76	27.77	26.00	27.95
Gate	41.23	41.22	38.37	41.76

Table 3.10: PSNR in [dB] for the different algorithms with equal noise in R, G and B channel, thus without the white balance step, $\sigma_n = 20$.

Sequence	Std. HT	Camera HT	Std. Bivariate	Camera Bivariate
Siegestor	28.01	28.21	21.71	25.93
Odeonsplatz	30.73	30.67	26.46	30.18
Ball Pool	30.28	29.90	26.13	29.79
Tree & Lights	25.14	25.19	21.38	23.94
Gate	38.20	38.10	31.89	37.25

obtain a better estimation of the optimal denoising result, we tried other metrics, which correlated better with human perception. However, the SSIM results, shown in Fig. 3.10, determine approximately the same maximum as the PSNR, and thus SSIM does not seem to help much in this application. The VIF shows the maximum quality for $k = 0$, which means for the noisy image. The VIF is hence not very helpful for choosing the correct denoising parameter. We therefore decided to use the PSNR in the following, and we will evaluate the results visually in the next section.

The PSNR results for the channel-independent noise, i.e., using noise gain factors that do not include the white balance, are shown in Tab. 3.9 and Tab. 3.10. Using our proposed noise gain factors in the wavelet hard thresholding shows similar results compared to the standard method, hence no significant improvement can be observed here. One reason could be that the SNR is more essential than the pure noise variance. As the signal variance is not taken into account in the hard thresholding denoising step, it seems to be not very advantageous to use a more correct noise variance. While the overall PSNR results of bivariate thresholding are not significantly higher than the hard thresholding results, it must be mentioned that the hard thresholding result shows severe blocking artifacts.

Bivariate thresholding takes into account the signal variance and applies a thresholding function which depends on the parent coefficient and thereby provides better denoising results than hard thresholding. This method can clearly be enhanced using our proposed more realistic noise variance. The proposed method shows a higher

Table 3.11: PSNR in [dB] for the different algorithms including the white balance step, $\sigma_n = 10$.

Sequence	Std. HT	Camera HT	Std. Bivariate	Camera Bivariate
Siegestor	27.78	27.81	24.43	28.09
Odeonsplatz	32.59	32.88	29.72	33.35
Ball Pool	31.85	31.49	28.93	32.26
Tree & Lights	25.24	25.45	23.22	25.58
Gate	39.67	39.83	34.64	39.57

Table 3.12: PSNR in [dB] for the different algorithms including the white balance step, $\sigma_n = 20$.

Sequence	Std. HT	Camera HT	Std. Bivariate	Camera Bivariate
Siegestor	22.83	22.90	19.63	22.98
Odeonsplatz	29.29	29.47	24.84	29.68
Ball Pool	27.88	27.67	23.98	28.15
Tree & Lights	21.37	21.63	18.88	21.40
Gate	35.10	36.90	28.86	35.01

PSNR for all the images and both tested noise levels. The improvement compared to the standard bivariate denoising is up to five dB and thus significant.

Including the white balance enlarges the improvement in terms of PSNR. For the channel-dependent noise, so including the white balance in our method, the camera-adapted method reaches up to 6 dB better results (Tab. 3.11 and Tab. 3.12).

Visual quality

While the hard thresholding results were comparable to the bivariate approach in terms of PSNR, the visual quality is inadequate when the threshold is set to achieve the optimal PSNR value. We thus conclude that the hard thresholding can be used for moderate denoising and desist from showing image examples in this dissertation. The bivariate denoising shows visually much better results compared to hard thresholding. We selected a representative image crop to compare the standard and the proposed method using a three- or five-level decomposition.

Fig. 3.9 shows the noisy image with four different denoising results. Both with the three-level (a and d) and with the five-level decomposition (b and e) the proposed method reduces more noise while retaining the details in the image. As the five-level-decomposition result in Fig. 3.9 (e) shows the overall highest quality, we conclude that the additional buffer requirements should be worthwhile.

3.2.4 Conclusion

This section first describes a simulation procedure to obtain noise gain factors, which allow us to capture the noise characteristics of spatially correlated camera noise efficiently in look-up-tables. The proposed adapted version of the bivariate wavelet denoising method accounts for the properties of real camera data and can operate in real time on an FPGA.

Compared to the original bivariate wavelet denoising approach using the standard median estimator for the noise variance, an improvement of up to 6 dB is achieved. Visual comparison shows that it reduces more noise and preserves more details. Our results hence demonstrate the importance of accounting for realistic camera noise characteristics in denoising approaches. We expect that a comparable performance increase might be achievable in other denoising methods that are based on the signal and the noise variance.

3.3 Comparison

The test sequences used in Chapter 2.3 are the basis for comparing the two methods presented in this chapter, the SA-DCT Bayer denoising method (SA-DCT Bayer), presented in Sec. 3.1, and the bivariate wavelet denoising based on noise gain factors (wavelet NGF), presented in Sec. 3.2. The SA-DCT Bayer uses a joint shape selection based on an estimated luminance image, and the denoising step is performed independently in the four subchannels of the Bayer data. The SA-DCT Bayer includes the signal-dependent camera noise model for the sensor data, described in Sec. 2.2.1. Conversely, the bivariate wavelet denoising based on noise gain factors (wavelet NGF) is applied on linear RGB data, which has been debayered before, and hence, it includes the spatial correlation of the camera noise after debayering.

Visual quality and PSNR

The SA-DCT denoises well in smooth areas and along edges and leads to a very natural appearance. No artifacts or artificial structures of the DCT are visible and the noise characteristic remains almost unchanged, due to the flexible shapes and the averaging of overlapping shapes. In contrast, the remaining noise of the wavelet NGF method is not as natural as the original noise, since the thresholding completely deletes coefficients, and thus the noise can appear blocky. The wavelet NGF method, however, reduces the noise more than the SA-DCT. Overall, both methods show visually appealing results.

The PSNR values on the city sequence, presented in Tab. 3.13, also indicate that the wavelet NGF method reduces more noise: the PSNR is increased by 2.4 dB with the wavelet NGF and by 1 dB with the SA-DCT Bayer. The SA-DCT Bayer shows a reliable improvement of around 1 dB for all sequences. With the wavelet NGF method, in contrast, the PSNR values of the real camera sequences are not always increased. The PSNR on the candy and the ball pool sequence are even lower than the PSNR of the noisy image. This is a surprise since higher PSNR is usually associated with less noise and thus higher image quality, but the visual comparison shows that the wavelet NGF reduces noise well and improves visual

Table 3.13: PSNR results in [dB] of the SA-DCT Bayer denoising method and the proposed bivariate wavelet denoising adapted to camera noise using noise gain factors (wavelet NGF).

	ball pool	candy	city	street	arch
Noisy	30.86	30.14	35.91	37.04	32.68
SA-DCT Bayer	**31.54**	**31.14**	36.66	**37.90**	**33.78**
Wavelet NGF	29.27	27.40	**38.30**	37.61	33.42

quality significantly. The question is thus, why are the PSNR values not showing the improvement on these two camera sequences?

How the crosstalk effect influences the evaluation of denoising results

The reason for the low PSNR values of the wavelet NGF results is the crosstalk effect. In image sensors, crosstalk describes that a pixel value does not only depend on the amount of light of one color band at the respective location, but is influenced additionally by the neighboring pixel values. A main reason for that is the crosstalk of the read-out circuit, and in sensors with a Bayer pattern, this causes that the effect is color-dependent. The crosstalk is clearly visible in highly saturated areas, e.g., a red area, where the blue signal is low but the red signal is high: high red values lead to higher green values for pixels in the same line, whereas the green values in the line with the blue pixels are lower. The different green values lead to a regular pattern in the final image, which can reduce visual quality significantly. Therefore, crosstalk compensation methods have been proposed, e.g., by Andriani et al. in [15]. Since crosstalk compensation is not included in our evaluation, the crosstalk effect is visible in the noisy images, and since it is visible in every frame, it is also visible in the mean image that is used as a reference. The SA-DCT method denoises the four subchannels independently, and hence, it does not change the crosstalk. The wavelet NGF method, in contrast, reduces the noise in the RGB data, and therewith it also reduces the crosstalk pattern. The bivariate method reduces coefficients that are not protected by a high parent coefficient, and therefore, it is particularly effective in reducing crosstalk patterns because the crosstalk only affects the first wavelet level.

The above explained effect is visible when comparing the respective difference images of the noisy and denoised images to the mean image. Fig. 3.11 shows the selected crop of the candy sequence that is used for illustration. The difference images shown in Fig. 3.11b - 3.11d are multiplied by 3 for illustration. The crosstalk effect is not visible in the difference of the reference to the noisy image, because it is equally strong in both. The crosstalk pattern is not visible in the SA-DCT Bayer difference image (Fig. 3.11c), since the SA-DCT processes the Bayer subchannels independently, and hence it does not change the crosstalk. The pattern is, however, clearly visible in the difference image of the wavelet NGF as it reduces the crosstalk together with reducing noise, see the bright sweets in the lower right of Fig.3.11d.

We conclude that the advantage of the wavelet NGF method is that it reduces more noise and additionally reduces the crosstalk effect. Conversely, the SA-DCT result is perceived as more natural, because the noise characteristic changes less than with the wavelet NGF.

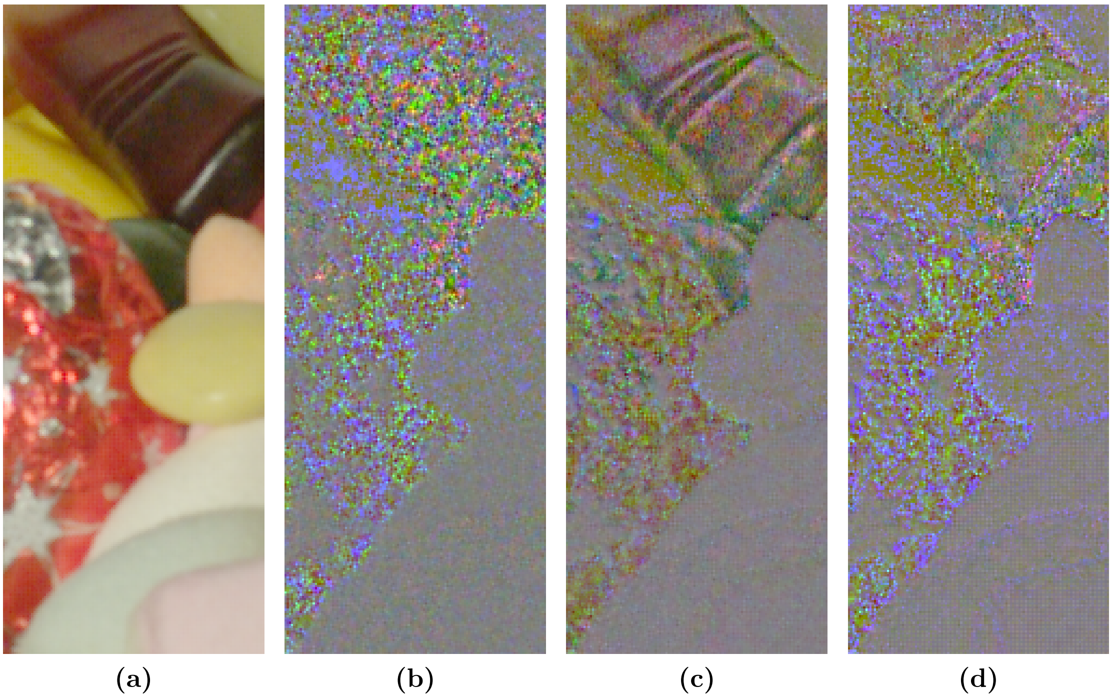


Figure 3.11: (a) shows a crop of the reference image for the test image candy that is used for comparison, (b) the difference of the noisy image to the reference, (c) the difference image for SA-DCT Bayer denoising, and (d)the difference image for the bivariate wavelet denoising based on noise gain factors.

Implementation aspects

Both proposed denoising methods are designed for very low computational complexity to enable real-time processing in embedded systems. The SA-DCT can be implemented similarly to the standard block DCT, and as most compression algorithms use a DCT, efficient hardware implementations are already available. Since the proposed SA-DCT operates on Bayer data, which has only one value per pixel, only one third of the RGB data needs to be processed, a clear advantage compared to RGB denoising methods. As shapes can overlap, memory capacities for buffering the estimates are needed. The buffer size scales with the required neighborhood, the window size, and hence can be adjusted according to the available resources. However, the minimal window size is higher than for an RGB method as denoising is performed on the subchannels of the Bayer data, and therefore, compared to RGB denoising, only one fourth of the pixel values of a region is available for denoising.

The wavelet NGF is based on the easy-to-compute shift-invariant DWT using low complexity filters. Sec. 3.2 shows that high-quality results can be achieved, despite the simple transformation. Despite the fact that further improvement certainly could be achieved with better filters or more complex transformations, e.g., a dual-tree or stationary transformation, we do not use these improvements, as the computational complexity would increase dramatically, and the wavelet decomposition accounts for the main part of the required resources. The resource requirements of the wavelet decomposition are scalable with the number of wavelet decomposition levels; in particular memory requirements can be adjusted with the number of decomposition levels. However, extremely low levels do not provide a significant increase in image quality for RGB data with spatially correlated camera noise; based on the evaluation in Sec. 3.2, it can be concluded that at least three levels should be used. A three level DWT was implemented on an FPGA and shows real-time performance on HD data with 30 fps. However, the resources needed for the wavelet decomposition might still be too extensive in camera systems that include a high amount of image processing modules. However, a wavelet decomposition can also be helpful for other image processing tasks, e.g., multiscale algorithms in computer vision, and in that case it might be worth the implementation effort.

For both methods, the available internal and external buffer influences the quality. To find a good trade-off between visual quality and memory requirements, the SA-DCT window size or, respectively, the number of decomposition levels can be adjusted. In a stream-based processing pipeline, window size is critical, because the pixel values are stored linewise. This can lead to extremely high buffer requirements for large windows and sensors with high resolution. Reordering the pixel values to a blockwise transmission could therefore reduce memory requirements significantly in the denoising step. The required buffer to reorder the pixels would of course compensate for the buffers saved in the denoising step, but if other image processing tasks also require a neighborhood the reordering can help.

Combining the spatial denoising with a temporal denoising step significantly improves the results of the SA-DCT Bayer method and a similar improvement can be expected for other spatial denoising methods. Hence, if an additional frame buffer is affordable in the system, a temporal denoising step should be included.

4 Parameter Estimation for Denoising

The aim of all denoising methods is to suppress the noise as much as possible while still retaining the fine details in the image. To find the optimal tradeoff, almost every method has one or more parameters that regulate the denoising strength. If the denoising strength is too low, too much noise remains in the image, whereas too strong denoising leads to oversmoothing of textures and small details and hence to a less natural image. The highest visual quality can be achieved only when the parameters are set to the optimal values. These values usually depend on both the image content and the noise in the image. The optimization of these parameters is crucial to achieve the best possible image quality in real applications.

The parameter setting is usually based on an image set that contains representative reference images, e.g., the well-known Kodak data set [136], with corresponding noisy images that are obtained adding artificial noise. Such an image set allows one to easily calculate denoising results with varying parameters. As subjective tests with human observers are often too costly and time-consuming, the denoising results are subsequently rated using an image quality metric, e.g., PSNR, to find the optimal denoising results and therewith the optimal parameters. The parameters as determined based on the test set are subsequently used for denoising all images in the real application.

This technique has two main drawbacks. First, the choice of the test set crucially influences how well the chosen parameters fit the real application. Finding a good test set is challenging; the need of a reference image limits the test data to unrealistic data, because in the real application the reference is not available – otherwise denoising wouldn't be necessary. Thus, the test set either contains some image data with artificial noise (e.g., the Kodak data set [136]) or artificial images with realistic noise (e.g., [10]). As this test data does not completely correspond to the data in the real world application, the parameter tuning on such test sets may lead to suboptimal results. Second, the parameters are determined using the test data and then are fixed in the real world application. This means that even the most carefully selected test set cannot lead to best results, because the image and noise statistics of the individual image in the application are not considered. To achieve optimal results in the real world application, an image-adaptive, automatic and referenceless parameter tuning method is needed.

Two different strategies for referenceless parameter estimation are known: on the one hand, analytical approaches for parameter optimization such as cross-validation [45], the L-curve [50] method, or Steins unbiased risk estimate (SURE, [102]) could be used. This proves to be computationally expensive and does not reflect the visual perception of images [132]. On the other hand, a "no-reference metric" (NR metric), i.e., a metric that can rate image quality without a reference, could be used. MetricQ [132] was specifically proposed as an estimator for denoising parameter tuning, but more general NR metrics can also be used, e.g., [70,87,88,90,91,105,106]. However, it is usually not possible to apply these metrics in the real world application, because they are too computationally complex to be calculated for every image.

We therefore propose a very efficient method with low computational complexity, which adapts to the actual image content using gradient histograms. As the human visual system is more sensitive to differences than to absolute values, gradient histograms are linked to the human perception more than absolute statistics. Bayesian denoising methods therefore use gradient distributions to improve their results, as in [31, 107]. In [133] the gradient histogram statistics were integrated in a sparse recovery algorithm, which led to better denoising results in textured areas. This motivates us to design our method based on gradient histogram statistics. However, in contrast to the usual approach of modeling the histograms via a parameterized distribution, we directly estimate a gradient histogram from the noisy image. Our method is thus image-adaptive and computationally efficient.

The method and the first results were presented in [4]. This chapter provides a detailed explanation of the method and additionally extends the results to a broader range of denoising algorithms. Sec. 4.1.1 first explains the background of our work – the gradient histograms of noisy and denoised images – and explains how our findings help to detect the optimal denoising strength. In Sec. 4.1.2, we explain our method for estimating the reference gradient histogram based on the noisy image. To evaluate our approach, we conducted a visual test with 20 participants based on images denoised with the state-of-the-art algorithm BM3D. We explain the test setup and discuss the results in Sec. 4.2. Sec. 4.3 evaluates how our method performs with 6 other denoising methods. Finally, we conclude in Sec. 4.4.

4.1 Proposed Parameter Estimation Method

4.1.1 Gradient Histograms of noisy and denoised images

Figure 4.1 shows the gradient histogram of a noisy image as well as the gradient histogram of the corresponding noiseless reference image and the gradient histograms of three differently denoised images. As natural images in general contain many homogeneous areas with just a little change in intensity, the reference gradient histogram shows its maximum in the small image gradients close to zero. Image noise introduces intensity changes in those former homogeneous regions and thus influences the distribution of image gradients. Thus, the noisy image contains less gradients

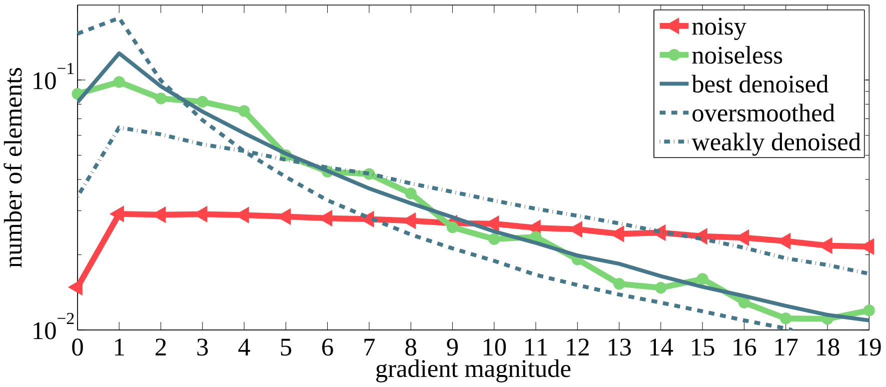


Figure 4.1: Gradient histograms for an image of the Kodak data set. We used zero mean additive white Gaussian noise with variance 0.01 to obtain the noisy image.

close to zero and the corresponding gradient histogram has less entries for the image gradients from zero to nine (see Fig. 4.1).

When denoising is applied, the gradient histogram becomes more similar to the reference gradient histogram again. The number of small gradients increases with increasing denoising strength. With further increasing denoising strength, the gradient distributions begin to diverge again and the image becomes over-smoothed. We therefore can detect whether the denoised image is over-smoothed or still noisy. Although we only use one image to illustrate the above effects in Figure 4.1, we observed them for almost every image in our data set. Based on our tests with BM3D applied to the Kodak data set, we suppose that the denoising strength is optimal when the corresponding gradient histogram matches the reference gradient histogram best.

We thus can automatically determine the optimal denoising strength by comparing the gradient histograms of differently denoised images with the corresponding reference gradient distribution. As the reference gradient distribution is not available in the real application, it has to be estimated. Hence, we discuss how to estimate the reference gradient distribution in the following section.

4.1.2 Estimating the gradient histogram

In literature, the reference gradient distribution is usually approximated using specific parameterized functions ([54]). Zuo et al. ([133]) use a hyper-Laplacian function as model and estimate the reference gradient distribution of an image using a one dimensional deconvolution model. Cho et al. ([32]) fit a generalized Gaussian distribution to the gradient histogram of a deconvolved and downsampled corrupted image to estimate the reference gradient distribution. However, using a model to describe the distribution of image gradients has two major drawbacks. First, the parameterization of a model might be complex. Second, the model might not be flexible enough to adapt to the gradient histogram of a specific image. Thus we estimate the gradient histogram directly from a downsampled image without using parameterization. This significantly reduces computational cost and does not limit the gradient histogram estimation to a predefined distribution.

Our first tests with the usual low resolution Kodak images provide proper estimates when our method is applied on the downsampled image. However, we don't obtain a good estimation of the reference gradient histogram when using the high resolution Kodak images or real camera data. This is due to the different image characteristics of the small Kodak data, which was already discussed in [11]. To obtain a better estimate for today's high resolution data, we have to upsample the downsampled image again and use the corresponding gradient histogram as reference distribution. Thus, our method consists of five steps:

1. Downsample the original noisy image
2. Upsample the downsampled image
3. Calculate the gradient histogram of the upsampled image
4. Calculate the gradient histogram of the denoised image
5. Compare the gradient histograms

The steps 4) and 5) are iterated for different denoising strengths until the minimal distance is found. We tested different combinations of algorithms for image downsampling and upsampling. One promising approach uses bilinear interpolation for downsampling by a factor of two followed by a DCT-based image upsampling to the original image size. Inspired by [30], the resistor-average distance (D_{RAD}, [59]) was used to measure the difference between the estimated reference histogram (Q) and the histogram of the denoised image (P). The resistor-average distance is based on the Kullback-Leibler divergence (D_{KLD}) as follows:

$$D_{KLD}(P \parallel Q) = \sum_{x \in X} P(x) \cdot log \frac{P(x)}{Q(x)} \tag{4.1}$$

$$D_{RAD}(P \parallel Q) = ((D_{KLD}(P \parallel Q))^{-1} + (D_{KLD}(Q \parallel P))^{-1})^{-1} \tag{4.2}$$

As our estimated reference gradient histogram is most similar to the true reference for small image gradients, we only use image gradients x with a range X from 2 to 10.

Matlab Code

To show how easy the method is to use, we present here the Matlab source code, which can be used to reproduce the results.

```
function hist_hg= getHist(img)
  img=im2double(img);
  hg=img(1:size(img,1)-1,:)-img(2:size(img,1),:);
  hist_hg = imhist (abs(hg));
  hist_hg = hist_hg' ./ sum(hist_hg(:));
end

function sig_opt=findSigOpt(img_noisy,img_denoised)

  % Step 1: Downsampling
  img_ds=imresize(img_noisy,0.5,'bilinear');
  % Step 2: Upsampling
  img_dct=dct2(im2double(img_ds));
  img_dct=padarray(img_dct,size(img_ds),0,'post');
  img_us=im2uint8(idct2(img_dct)*2);
  % Step 3: Calculate histogram of upsampled image
  hist_estimate=log(getHist(img_us));
  % Step 4+5: Calculate histogram of denoised image,
  % compare histograms
  for sigma=1:1:50
    hist_denoised=log(getHist(img_denoised{sigma}));
    rad_value(sigma)=RAD(hist_denoised(2:10),...
    hist_estimate(2:10));
  end
  [~,best_sigma]=min(rad_value);
end
```

Listing 4.1: This Matlab code performs the five steps from section 4.1.2. The input parameter img_noisy is a single noisy image and the input parameter img_denoised is a struct that contains several differently denoised images.

4.2 Evaluating the DUG metric applied to BM3D parameter estimation

To evaluate our method we need to compare the estimated parameters to the truly optimal parameters. The truly optimal parameters are the parameters leading to the denoising result with the best visual quality. As none of the usual quality metrics can reflect the human perception of visual quality perfectly, we can not use any objective metrics as a reference for our evaluation. As the visual quality can only be

Table 4.1: Error and mean execution time of different methods for estimating the BM3D parameter sigma. The error is the mean deviation of the parameter chosen by the respective estimation method to the optimal parameter determined based on the subjective test. The execution time is measured using the given Matlab implementations provided by the authors.

	NIQE	**BRISQUE**	**MetricQ**	**DIIVINE**	**BIQI**	**DUG**
Overall Mean Error	21	3.46	7.42	3.30	1.82	1.50
Kodak Mean Error	22.53	1.69	8.80	1.36	1.14	0.92
ARRI Mean Error	17.86	8	3.85	8.28	3.57	3.00
Execution Time (s)	0.540	0.223	0.207	25.338	0.517	0.014

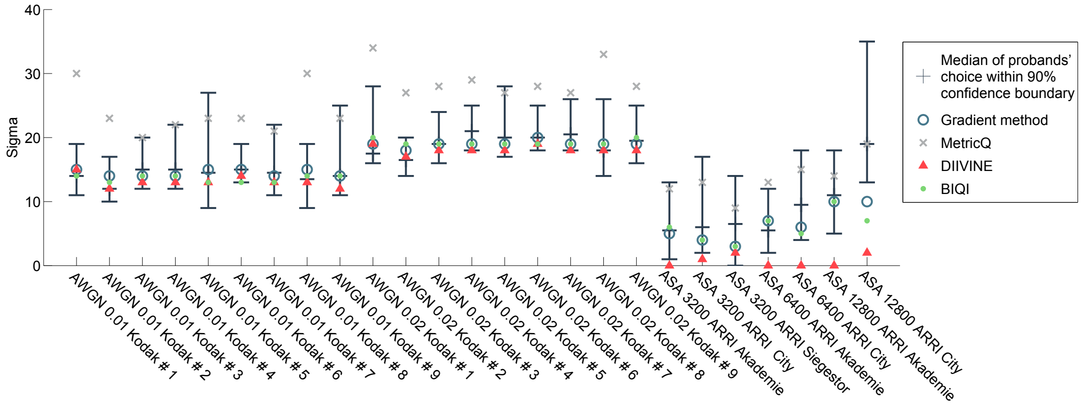


Figure 4.2: Evaluation results of the subjective test and the metrics for parameter estimation. The goal is to find the best sigma parameter for the BM3D denoising.

determined by human perception, we have to perform a subjective test with human observers that rate the visual quality and thus tell us the truly optimal parameters.

4.2.1 Test Data

We use the 9 test images of the Kodak data set [136] with a resolution of 768 × 512 pixels and added white Gaussian noise (AWGN) with a variance of 1% and 2%, respectively, on all color channels. Afterwards, we converted the color images to grayscale images by calculating the luminance value of each pixel. Since Andriani et.al. showed the limitations of this data set [11], we also included images from the ARRI data set [11] as well as simulated data with realistic camera noise [10] in the test. These images are taken with a digital camera, therefore no reference is available. The data, however, is video data and thus we can calculate a reference by using the mean of a static sequence. We thus include 7 images with realistic camera noise in our test. We apply the state-of-the-art denoising algorithm, BM3D [33], to all images. The denoising parameter σ regulates the denoising strength: a higher σ value leads to stronger denoising.

4.2.2 Subjective Test Setup

In the subjective test, we have asked a group of volunteers to select the optimal denoising parameter σ for each image in our data set, such that the noise in the image is reduced while still preserving the fine details in the image.

To this end, we developed an application in which a test subject could choose the denoising strength via a slider that could be controlled with two keyboard keys (”higher” or ”lower”) from a range of $\sigma \in [0.50]$ in steps of $\Delta\sigma = 1$. All images were pre-calculated such that a change of the parameter had an immediate effect on the displayed image. The participants were free in their choice of the parameter value and were given no time restriction for the setup, however, most subjects found their subjectively optimal denoising parameter for each image very quickly with very few iterations necessary.

To avoid any artifacts due to scaling of the images, all images were displayed in their native resolution (1:1 pixel mapping) on a calibrated reference studio monitor Sony BVM-L230. For the high resolution ARRI dataset, we cropped the images to a size of 1280x720 pixels to fit the monitor without downsampling. In accordance with [137], the distance between the observer and the screen was set up to be three times the height of the screen. The test was conducted at the video laboratory at Technische Universität München in a room compliant with ITU Rec. BT.500 [137]. Prior to the test, all participants were screened for visual acuity and color blindness. The group of participants consisted of mostly university students with no special background in image processing. In total, 20 test subjects, 9 female and 11 male, aged 19 to 30 with a mean age of 23 participated in the test. The results in Fig. 4.2 show the median σ value for each of the 25 test images and the respective confidence interval.

4.2.3 Test Results

Once we obtained the optimal parameters based on the subjective test, we can evaluate our method using the mean error. Our method shows an overall mean error of 1.5, which can be considered being very close to optimal, as the step size of the paramter in the subjective test was equal to one.

We compare the parameters found by our method to the parameter estimation using NR metrics. Tab. 4.1 shows the mean error of the estimated parameters with our method, compared to a parameter estimation using MetricQ [132] and recently proposed general NR metrics: the “blind image quality index” (BIQI) [91], the “distortion identification-based image verity and integrity (DIIVINE)” [90], the “blind/referenceless image spatial quality evaluator” (BRISQUE) [87] and “natural image quality evaluator" (NIQE) [88]. Our gradient based method DUG shows a lower error and thus matches better the visually best denoising parameters than the other methods. BIQI provides second best results. MetricQ shows a high error in the test, which is surprising as it was designed for denoising parameter selection specifically. We understand the reason by observing Fig. 4.2, which shows the test results for each test image. MetricQ choses too strong denoising for all our test images.

We also compare the runtime of our algorithm in a simple Matlab implementation with other NR metrics. Tab. 4.1 shows the corresponding mean execution times per image in seconds. We used the available Matlab implementations of the NR metrics for comparison. The measurements were performed on a 64-bit Windows System with Intel-Core i3-2100 and 8GB of RAM. Our method has the lowest execution time and is more than ten times faster than the fastest NR metric in comparison.

The results indicate that our method, while being very simple, can accurately estimate denoising parameters. In our subjective test with BM3D denoising, our method outperforms the other NR metrics.

4.3 Metric-based evaluation of the parameter estimation for different denoising methods

We proposed a both cost-efficient and accurate method for denoising parameter optimization. This section focuses on further tests including a broader range of denoising algorithms. As the subjective tests are extremely time consuming, we need a different method for evaluating a wider range of denoising algorithms. Despite the fact that the existing objective quality metrics are not perfect, full-reference metrics can be used for a basic evaluation. We present a comparison to two metrics: the PSNR, as it is widely used in denoising, and the IFC, as it has correlated best with human vision in tests with real camera noise [8]. We thus obtain two different reference parameters considered to be optimal, one chosen by PSNR optimization and one chosen by IFC optimization. In Tab. 4.2 to Tab. 4.8 the results are presented. The tables show the mean deviation of the parameter to the optimal parameter chosen by PSNR/IFC for Kodak data and ARRI data.

4.3.1 BM3D

The denoising method BM3D has already been used in the subjective test. Hence, the results in Tab. 4.2 allow us to compare the metric-based evaluation to the subjective-test-based evaluation (results in Tab. 4.1). The mean deviation of the estimated parameter to the PSNR- and IFC-optimized parameter is larger than the mean deviation of the estimated parameter to the subjective test parameter. This applies to the metrics DUG, BIQI and DIIVINE. In contrast, the deviation of MetricQ is smaller than in the subjective-test-based evaluation. This behavior can easily be explained: both PSNR and IFC overestimate the denoising strength, and from the last section we know that MetricQ shows the same behavior. However, despite the fact that the metric-based evaluation is not optimal for representing human perception, it provides us results that indicate how well the DUG method performs for other denoising methods. For BM3D, the DUG method reaches lowest deviation to PSNR and IFC, and this matches the human rating.

Table 4.2: Mean deviation of the parameter σ in BM3D denoising

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	3.60	5.80	6.00	4.04
Mean deviation (Kodak) PSNR	3.17	6.56	4.28	3.50
Mean deviation (ARRI) PSNR	4.71	3.86	10.43	5.43
Mean deviation IFC	3.24	6.00	5.40	3.68
Mean deviation (Kodak) IFC	3.50	6.22	4.61	3.83
Mean deviation (ARRI) IFC	2.57	5.43	7.43	3.29

4.3.2 BLS-GSM

BLS-GSM [101] is a global denoising method operating in wavelet domain. The authors provide a matlab package, which is used here. The parameter to adjust denoising strength is the noise standard deviation 'sigma'. All other parameters were set to the values suggested by the authors. Sigma was changed with a step size of 0.005 in the range of 0 to 0.25.

For the BLS-GSM parameter estimation, Tab. 4.3 shows the smallest error using MetricQ. However, DUG results are the second best method for BLS-GSM. As MetricQ usually overestimates the denoising stregth, results for DUG can be expected to be able to estimate the visual quality better than MetricQ.

Table 4.3: Mean deviation of the parameter σ in BLS-GSM denoising

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	1.84	1.24	3.00	2.96
Mean deviation (Kodak) PSNR	2.06	1.22	2.56	2.50
Mean deviation (ARRI) PSNR	1.29	1.29	4.14	4.14
Mean deviation IFC	1.56	1.12	2.48	2.84
Mean deviation (Kodak) IFC	1.89	1.39	2.39	2.11
Mean deviation (ARRI) IFC	0.71	0.43	2.71	4.71

4.3.3 Total Variation

Rudin, Osher and Fatemi [104] introduce the Total Variation (TV) as a regularizer which retains sharp edges in the image. Many improvements have been proposed, but in this test the classical implementation is used. The denoising strength is regulated by the parameter ν, which is adjusted from 0 to 50 with a step size of 1 for the Kodak data, and 0 to 200 with a step size of 5 for the ARRI data. We included this classic method because the artifacts are very different from the usual blur: typical are staircasing artifacts, which could lead to a different behavior in the gradient histograms.

The results in Tab. 4.4 show that BIQI and DUG provide the best and the second best result. That indicates that DUC is able to cope with different types of artifacts and is not limited to detect blur.

Table 4.4: Mean deviation of the parameter ν in Total Variation (TV) denoising

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	5.28	10.12	12.56	4.60
Mean deviation (Kodak) PSNR	4.33	9.56	7.22	3.89
Mean deviation (ARRI) PSNR	7.71	11.57	26.29	6.43
Mean deviation IFC	7.24	8.88	10.04	6.88
Mean deviation (Kodak) IFC	5.56	8.78	8.00	5.11
Mean deviation (ARRI) IFC	11.57	9.14	15.29	11.43

4.3.4 Wavelet hard thresholding

A well-known method for denoising is hard thresholding in the wavelet domain. A dual-tree wavelet transformation [110] was used, with three levels of decomposition. Coefficients smaller than a threshold t were set to zero. Compared to the other evaluated methods this simple method leads to insufficient results. The poor quality of the output images renders it difficult to rate an optimal parameter. The threshold was set from 0 up to 0.5 with a step size of 0.01.

The PSNR prefers a high threshold, thus it probably over-estimates the denoising strength. The PSNR-based results show the lowest error for MetricQ, which seems reasonable as MetricQ usually overestimates the denoising strength, too. The IFC results, in contrast, rate the necessary denoising strength much lower, which leads to the result, that DUG shows the lowest error for the Kodak data. However, using the ARRI data the error of BIQI and DUG is extremely high. This is due to a second minimum, which occurs at very high threshold (between 0.4 and 0.5). As such high thresholds are unrealistic to provide good results; this can be easily avoided by choosing a more appropriate range of the threshold parameters. Thus we conclude that DUG can provide good estimation results also with wavelet hard thresholding if the parameter range is well adjusted.

Table 4.5: Mean deviation of the threshold parameter in wavelet hard thresholding (HT)

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	6.20	1.32	3.88	6.40
Mean deviation (Kodak) PSNR	3.78	1.22	3.00	3.22
Mean deviation (ARRI) PSNR	12.43	1.57	6.14	14.57
Mean deviation IFC	6.40	5.36	3.28	6.52
Mean deviation (Kodak) IFC	3.22	6.44	3.78	3.78
Mean deviation (ARRI) IFC	14.57	2.57	2.00	13.57

4.3.5 Gaussian filtering

The simplest method used in image denoising is probably Gaussian filtering. While it is not an option in high-quality applications, the method is very basic and shows the classical denoising artifact, it blurs the image. We use a filter kernel based on a 5×5 neighborhood and change the filter kernel size by adjusting σ from 0.1 to 5.1 with a step size of 0.1. For classical blurring the MetricQ seems to underestimate the denoising strength and thus provides the best results based on IFC. Compared to PSNR the DUG and BIQI perform very well.

Table 4.6: Mean deviation of the parameter σ, which adjusts the spatial kernel size of the gaussian filter

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	2.92	4.12	4.16	2.48
Mean deviation (Kodak) PSNR	2.00	3.67	2.33	1.83
Mean deviation (ARRI) PSNR	5.29	5.29	8.86	4.14
Mean deviation IFC	3.08	2.52	4.64	3.20
Mean deviation (Kodak) IFC	2.94	2.17	3.39	3.00
Mean deviation (ARRI) IFC	3.43	3.43	7.86	3.71

4.3.6 Bilateral filtering

Bilateral filters and derivations are widely used in low cost denoising. We use a spatial kernel with $\sigma_{spat} = 3$ and adjust the range kernel σ_{range} from 0 to 1600 with a step size of 250 (for 16 bit values). The window size is 9×9.

The results in Tab. 4.7 show the lowest error for MetricQ in the IFC based evaluation. We observed that the denoising parameter was overestimated by all other

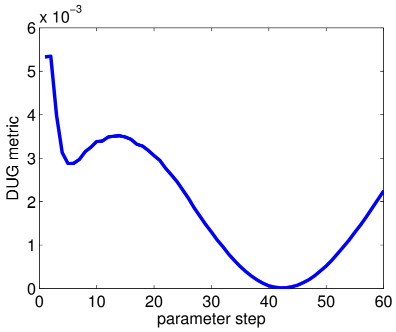


Figure 4.3: The graph shows the DUG metric result for bilateral filtering. The x-axes corresponds to the step of the range kernel parameter σ_{range} (0 to 1600 in 16 bit values).

methods, thus PSNR, BIQI, DIIVINE and also DUG. Only IFC and MetricQ seem to provide good results. For DUG, this behavior comes with multiple minima: the first minimum in Fig. 4.3 would probably be at the correct position, however, another minimum occurs at higher denoising strength and this leads to the wrong parameter choice. As mentioned before, this can be avoided by selecting an appropriate parameter range based on test data.

Table 4.7: Mean deviation of the parameter σ_{range}, which adjusts the range kernel in the bilateral filter

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	9.60	17.64	12.12	5.24
Mean deviation (Kodak) PSNR	9.22	21.94	12.17	5.22
Mean deviation (ARRI) PSNR	10.57	6.57	12.00	5.29
Mean deviation IFC	16.80	7.96	18.68	11.88
Mean deviation (Kodak) IFC	21.61	10.00	24.56	14.17
Mean deviation (ARRI) IFC	4.43	2.71	3.57	6.00

4.3.7 DCT-based hard thresholding

A sliding window DCT hard thresholding was implemented. The blocks of 8 × 8 size are transformed in DCT domain. Blocks are selected every third pixel, thus they are overlapping. Hard thresholding is applied in the DCT domain and the overlapping results are averaged. The threshold range is 0 to 1 with a step size of 0.02. The results of the DUG metric are better than the results of BIQI, MetricQ and DIIVINE for the Kodak data in the PSNR-based evaluation. In the IFC-based evaluation on the Kodak data BIQI and DIIVINE provide best results, and DUG follows them closely. However, the DUG metric generates wrong results for the

ARRI data, which leads to a high overall error. Similar to the above results, this seems to be due to double minima. Further research would be necessary to optimize the DUG metric for ARRI data.

Table 4.8: Mean deviation of the threshold parameter in a sliding-window block DCT denoising using hard thresholding

	DUG	MetricQ	DIIVINE	BIQI
Mean deviation PSNR	7.20	1.60	3.32	1.96
Mean deviation (Kodak) PSNR	1.11	1.72	1.56	1.72
Mean deviation (ARRI) PSNR	22.86	1.29	7.86	2.57
Mean deviation IFC	8.12	3.40	2.96	1.36
Mean deviation (Kodak) IFC	2.67	4.39	1.44	1.50
Mean deviation (ARRI) IFC	22.14	0.86	6.86	1.00

4.4 Conclusion

We proposed an image-adaptive method to automatically determine denoising parameters without a reference. The method is based on gradient histogram comparison. We showed that the optimal denoising strength can be determined by comparing the gradient histogram of the denoised image to the true gradient histogram. To estimate this true gradient histogram, we propose an image-adaptive approach, which is solely based on the noisy image. First, downsampling is applied to the noisy image, second it is upsampled to its original size, and finally the gradient histogram is calculated on the upsampled image. The method is very fast and easy to calculate. To evaluate the accuracy of our approach, we conduct a subjective test with 20 participants. In our test we included images from both the well-known Kodak data set and the ARRI data set, a set of realistic camera data. We used the state-of-the-art method BM3D for denoising. The test results provide us a reference for the optimal parameter, which can be compared to our method, to MetricQ, and to four general NR metrics. Our method provides parameter estimation results that lead to higher visual quality and significantly lower execution times than all other tested metrics. We further evaluated the performance of the DUG metric based on an evaluation taking the PSNR and IFC as a reference metric. While this evaluation is not as reliable as the subjective test, it can still provide an indication for the performance of DUG with other denoising methods. The results show that the DUG metric works well in most cases. However, in some cases either BIQI or MetricQ seem to perform better. Furthermore, it was shown that the parameter range must be adjusted appropriately before the test, as otherwise double minima can lead to wrong parameter estimation results using DUG. There is room for improvement for the ARRI data. Given an appropriate parameter range setting, the overall the performance of DUG is definitely acceptable, especially given its extremely low implementation cost.

5 Quality Evaluation

Rating the quality of the denoising results is an important task to test, optimize, and develop denoising algorithms. Denoising algorithms can be tested automatically and optimized efficiently only with a realistic test setup. Whereas the last chapter proposes a method for the optimization of denoising parameters, this chapter presents a more general approach for the evaluation of denoising results and an analysis of the underlying aspects of human vision.

The first section in this chapter, Sec. 5.1, proposes a method to evaluate denoising methods more realistically, which is based on previous publications [8, 10]. Whereas AWGN allows generating test data easily in a repeatable manner, it does not reflect the noise characteristics of a real digital camera. Chapter 2 concludes that realistic camera noise is fundamentally different from AWGN. Using such unrealistic data to test, optimize, and compare denoising algorithms may lead to incorrect parameter tuning or suboptimal choices in research on denoising algorithms. In the following section, we therefore propose an approach to evaluate denoising algorithms with respect to realistic camera noise, including a new camera noise model and the full processing chain of a single sensor camera. Based on that model, the visual quality of noisy and denoised test sequences is determined in a subjective test with 18 participants. The results show that the noise characteristics have a significant effect on visual quality. We apply and evaluate the performance of 12 full-reference metrics. As no-reference metrics are especially useful for parameter tuning and evaluation of real camera sequences, we additionally evaluate five no-reference metrics with our realistic test data.

As the first section of this chapter shows the significant effect of the noise characteristics on the quality metric performance for real camera noise and spatially denoised video sequences, improving quality metrics is necessary. To that end, Sec. 5.2 discusses the aspects of human vision that might explain the human perception of noise. As noise in camera data is not white, it has frequency-dependent noise characteristics. If sequences are additionally processed with a spatial denoising, the frequency-dependence becomes even stronger as high-frequency noise is reduced. Compression could lead to a similar effect. The question is how this frequency-dependence affects the perceived image quality. Since human vision is clearly frequency-dependent, we analyze noise visibility with respect to different frequency bands. Most quality metrics are designed for single images; however, this dissertation tackles video denoising. Therefore, the visibility of static (spatial) and dynamic (spatiotemporal) noise is compared.

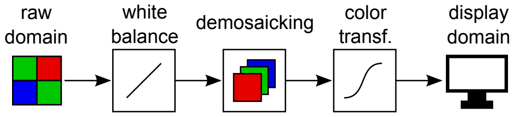


Figure 5.1: Image processing pipeline of a single-sensor camera.

5.1 Evaluating Denoising Algorithms with Respect to Realistic Camera Noise

The development and tuning of denoising algorithms is typically based upon standard test datasets like the Kodak image set [136]. These datasets include a collection of representative reference images. To evaluate denoising algorithms, the reference images are degraded using artificial noise, to obtain pairs of a reference and a noisy image. In the simplest case, the results of the denoising algorithms are subsequently evaluated using the difference to the reference image (PSNR). As this measure does not correlate well with human perception of visual quality, several more sophisticated quality metrics have been proposed [24, 35, 99, 112, 124–126, 132].

Typically, the mentioned noisy images are generated by applying additive white Gaussian noise (AWGN) to the reference images. While AWGN allows us to easily generate noisy images in a repeatable manner, it does not reflect the properties of realistic camera noise. In [121] it is shown that noise in the raw sensor data is signal dependent. While in Chapter 2 a detailed overview of the processing pipeline is given, we recall in Fig. 5.1 the three major steps which are required to transform the raw sensor data to an image that can be viewed on a display device (display-domain image). The first step is the white balance. Since the sensor data only provides one color value per pixel (Bayer mask) debayering is required, as a second step, to obtain a full-color image. In a third step, a color transformation is applied to transform the image into the monitor color space. Previous work showed that the second step, the debayering, has a significant effect on denoising results using the Kodak data set, as it introduces a spatial correlation in the noise characteristics [97]. Further, the color transformation changes the noise distribution in a nonlinear manner. Hence, the realistic noise characteristic in the camera data is fundamentally different from AWGN.

Thus, the complete processing pipeline must be considered to generate test images with realistic camera noise. Some recent denoising methods implement more realistic camera noise models, e.g., [38, 43, 76]. To evaluate these algorithms representative result images based on real camera data are shown. To quantitatively compare their results simple models as AWGN are used. When denoising algorithms are evaluated and optimized using unrealistic test data, this, however, may lead to wrong parameter tuning or suboptimal choices in research and development of denoising algorithms.

In the following section, we propose an approach to evaluate denoising algorithms with respect to realistic camera noise. Following the first results in [10], [8] presents an extension of the work to blind image quality evaluation for denoising results. In this section, the approach is explained in detail.

We first describe a new camera noise model that includes the complete processing chain. Further, two state-of-the-art denoising algorithms are evaluated with respect to their denoising results on test images with AWGN and images generated using our new realistic camera noise model. To evaluate the visual quality of the denoised images a reliable quality metric is required. Up until now, quality metrics have not been tested with respect to realistic camera noise. Thus, to determine the visual quality of the test images in a reliable way, a subjective test with human observers is performed. Further, the results of the subjective test are compared to a large set of existing quality metrics. We analyze in detail how the individual processing steps influence the performance of these metrics. This allows us to identify the most suitable metrics to evaluate denoising algorithms with respect to realistic camera noise.

Blind quality evaluation, i.e., rating the visual quality without a reference image, is crucial for the automatic parameter tuning of denoising algorithms. Furthermore, these metrics would enable us to directly use test data without reference. Thus, real camera data could be used for denoising evaluation directly. So-called no-reference metrics (NR metrics) are therefore of high interest for the evaluation and parameter tuning of denoising algorithms. As recent research results on NR metrics are very promising [87, 88, 90, 105], we also discuss blind image quality evaluation in this section.

The remainder of this section is organized as follows: the test setup for the subsequent experiments including the subjective test is outlined in Sec. 5.1.1. We discuss the subjective quality results and we compare the performance of full-reference and no-reference quality metrics with respect to realistic camera noise in Sec. 5.1.2. In Sec. 5.1.3, two state-of-the-art denoising algorithms are evaluated using test images with AWGN and test images with the new realistic camera noise model and quality metrics are compared using the Spearman correlation over all the test sequences. Finally, we conclude in Sec. 5.1.4.

5.1.1 Test Setup

The images in display domain have a noise characteristic that is spatially correlated, signal-dependent and with an unknown noise distribution. Hence, the noise characteristic is very different to AWGN in the display domain, which is the adequate domain for human observers and thus the appropriate domain for the evaluation of denoising algorithms. For a realistic evaluation, test data in the display domain is required. We show how to obtain realistic test data taking into account the complete camera processing pipeline.

To evaluate denoising algorithms, our test setup consists of a reference image, a noisy image and a denoised image. While noisy images could be obtained in the form of real camera data, we would lack the corresponding reference imagery. Applying noise to a readily processed image, e.g., an image from the Kodak data set, cannot be used for tests with realistic camera noise, as it is necessary to include all the processing steps described in Sec. 2.1. To obtain realistic test data, a full-color reference in the linear domain is required. We obtained this reference data from a rendered 3D scene.

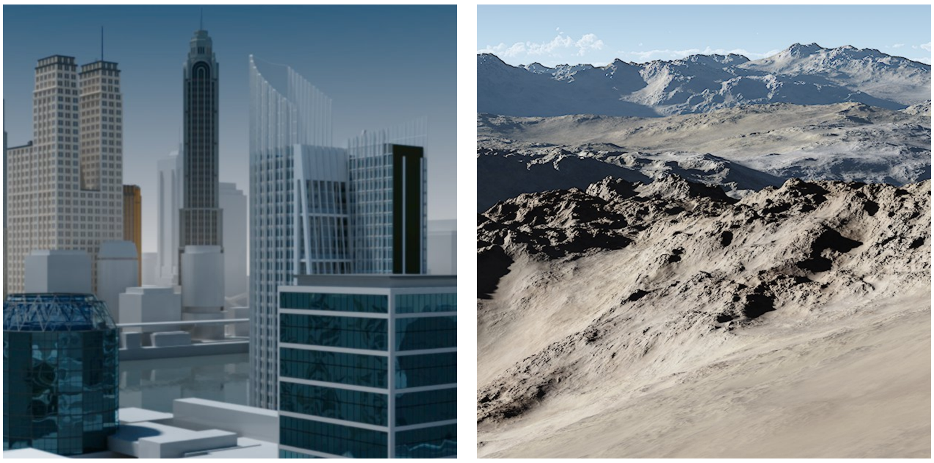
Reference sequences

For our test we use two different scenarios, one pan over a city, named “city”, Fig. 5.2a, and a landscape sequence obtained by moving a static image by a few pixels to provide a video sequence, named “landscape”, Fig. 5.2b. The sequences have been chosen to reflect typical challenges in denoising natural images. The city sequence is dominated by horizontal and vertical edges and squares, whereas the landscape sequence has a lot of fine details that are not part of larger structures. In the landscape sequence, the optical low pass filter of the simulated camera has been adjusted to be less restrictive such that more high-frequency content is left in the images. The sequences are 16 bit data rendered in linear domain. To incorporate the optics of a camera system the images are multiplied in the Fourier domain with the optical transfer function of the camera. This step takes into account the diffraction limited lens, the optical low pass filter and the pixel aperture. For details we refer to [109]. The rendering of the 3D scene has to provide high resolution images to avoid aliasing effects in the camera optics simulation. We use a resolution of 4096^2 pixels to obtain 1024^2-sized images as simulation output. This approach provides realistic reference data in linear domain. Applying the color transformation to this data provides a display-domain reference image.

Test sequences

Besides the reference images, noisy test images are required for denoising evaluation. To generate these noisy images, we use the most usual and most simple model, AWGN in display domain, and our new realistic camera noise model. Additionally we use noise models that enable us to evaluate the impact on visual quality of the two main differences between camera noise and AWGN individually: signal-dependence and spatial correlation through debayering.

In Fig. 5.3 the usual AWGN model corresponds to the simulation path named ”noisy AWGN uncorrelated“. We use AWGN with a standard deviation of 1100 in 16 bit domain, which is equivalent to approximately 1.7% of the signal range. To evaluate the influence of the debayering step on visual quality, we simulate AWGN and apply a Bayer mask with subsequent debayering, which is represented by the



(a) city **(b)** landscape

Figure 5.2: One frame of the computer-generated test sequence

simulation path ”noisy AWGN demosaicked“ in Fig. 5.3. While this does not correspond to real camera data, we can expect more realistic results by including this step into the usual AWGN model. To obtain signal-dependent noise, we replace the AWGN based noise model with the signal-dependent camera noise from Sec. 2.2.1 added in linear domain (“noisy SD uncorrelated” in Fig. 5.3). The noise level of the camera noise in our test sequences corresponds to a sensitivity of 3200 EI. To obtain realistic camera noise the debayering step must be included, thus the simulation path for realistic camera noise is ”noisy SD demosaicked“ in Fig. 5.3.

To obtain denoised sequences, all the noisy sequences from Fig. 5.3 are denoised with two different denoising algorithms, BM3D [33] and BLS-GSM [101].

Subjective test design

With the approach described above, we obtained reference, noisy and denoised images. To evaluate denoising algorithms, the visual quality of the noisy and the respective denoised image must be compared. We thus require a metric, which can reflect the visual quality of the noisy and denoised test images similar to the human perception. However, to the best of our knowledge, the performance of quality metrics has not been validated for realistic camera noise.

The validation of quality metrics is based on databases that contain test images with artificial degradations and respective results of subjective tests, in which the test participants assess the perceived visual quality of these test images. To determine quality metric performance, the metrics are applied to the database test images and the results are compared to the respective subjective test result. While databases as the LIVE [49], the TID2008 [100] and the CSIQ [65] database were used to evaluate state-of-the-art quality metric performance with respect to differ-

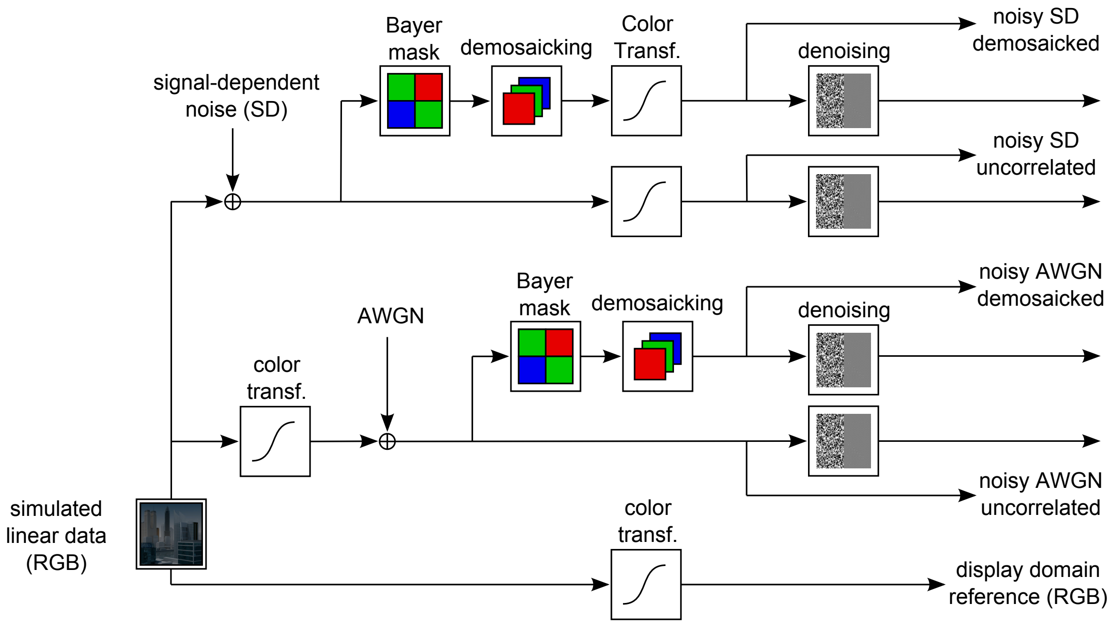


Figure 5.3: Processing of simulated sensor data for the test using signal-dependent noise (SD) and AWGN.

ent noise models, none of them uses realistic camera noise. The performance of the quality metrics is thus unknown for camera noise.

To obtain reliable information on the visual quality of our noisy and denoised test sequences, we conduct a subjective test with our test material. We used the double stimulus DSIS methodology with a undistorted reference and impaired noisy sequence according to ITU-R BT.500. A discrete scale from 1 to 10, representing a impairment range of “very annoying” to “imperceptible”, was used. The test participants were 18 students, aged between 20 and 30. The task for the participants was to assess the perceived impairment of the images. Before each test, a training session was performed and the expected distortion, blur and noise, were mentioned in this training. The test was performed in the ITU-R BT.500 compliant video quality evaluation laboratory at the Institute for Data Processing at Technische Universität München. For displaying the videos, a color calibrated Sony BVM-L230 reference LCD display with a screen diagonal of 23 inches was used. To get reliable results, the outlier were removed in the post processing of the subjects’ votes. Votes were removed, if they deviated more than 2σ from the mean for a sequence. Using this criterion, 4.6% of all votes were discarded. After outlier removal, the mean opinion score (MOS) was determined for the different test images.

The subjective test results provide reliable values for the subjective quality of all our test images. These results enable us to compare the visual quality of the noisy and denoised test sequences. Furthermore, the test data enables us to determine the performance of quality metrics with respect to realistic camera noise.

5.1.2 Visual quality of sequences degraded by camera noise

The performance of denoising algorithms is evaluated by comparing the quality of a denoised image to the quality of the respective noisy image. To determine the performance of denoising algorithms we therefore need to evaluate the quality of the noisy images first.

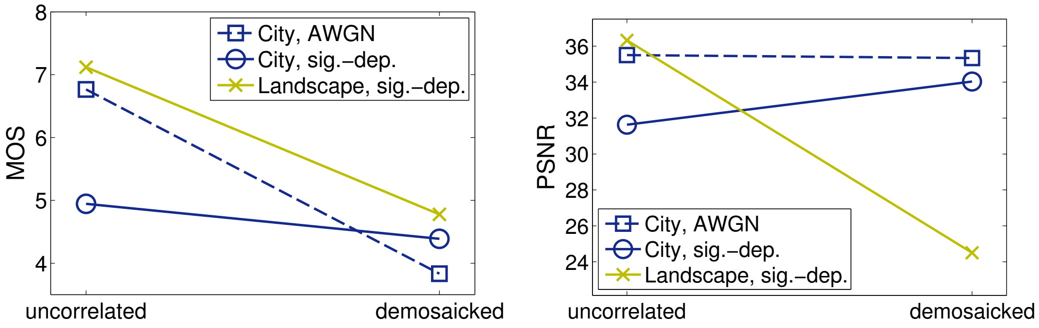
Subjective quality

As described in Sec 5.1.1, four different noise models were used in our test: the usual AWGN model, AWGN with debayering, signal-dependent noise without debayering and finally the realistic camera noise model – signal-dependent noise with debayering. In this section, we evaluate the visual quality of the noisy test sequences and analyze the main differences between the realistic camera noise and AWGN: spatial correlation introduced through debayering and signal-dependence.

In Fig. 5.4a the MOS results are shown for sequences without the debayering step, thus containing uncorrelated noise, on the left, and the results for noisy sequences with debayering are shown on the right. We notice that the debayering leads to a lower MOS for all three test sequences. The MOS of the city sequence with AWGN is about 3 scores lower when debayering is included. Regarding the sequences with signal-dependent noise, the MOS is 0.5 lower for the city sequence and 2.3 lower for the landscape sequence when debayering is included. The debayering changes the difference in visual quality of the sequences: while it is noticeable that the MOS for the uncorrelated noisy landscape sequence is quite good (7.1) compared to the respective city sequence (4.9), the difference is much smaller (0.4) when debayering is included. This may be explained by the image content: while the fine grain of the uncorrelated noise that is hardly differentiable to the sand, the coarser grain of the spatially correlated noise that is clearly visible in the landscape sequence. For all three test sequences the spatially correlated noise is perceived as more annoying than the uncorrelated noise. Furthermore, the debayering changes the difference in visual quality of different noisy images.

To reflect realistic camera noise a signal-dependent noise model is required. The noise level of signal-dependent noise depends on the signal and thus on the image content. Through the color transformation, the signal-dependence is nonlinear in display-domain images, usually noise is most visible in dark regions of the image. The noise level of signal-dependent noise is varying with the image content and thus not directly comparable to AWGN. For the city sequence we simulated both AWGN and the signal-dependent noise. While AWGN is classified as less annoying than signal-dependent noise when it is uncorrelated, AWGN with debayering is classified as more annoying than signal-dependent noise with debayering. That means, the relative order is changed with the noise model.

The MOS results showed that debayering has a significant impact on visual quality, in our test it leads to a lower visual quality of the noisy sequences. The noise model



(a) (b)

Figure 5.4: The MOS and PSNR results for the test sequences "City" and "Landscape" using the traditional AWGN model (dashed) and the realistic signal-dependent noise (solid lines). The uncorrelated noise, processed without debayering, is shown on the left, the results with debayering on the right.

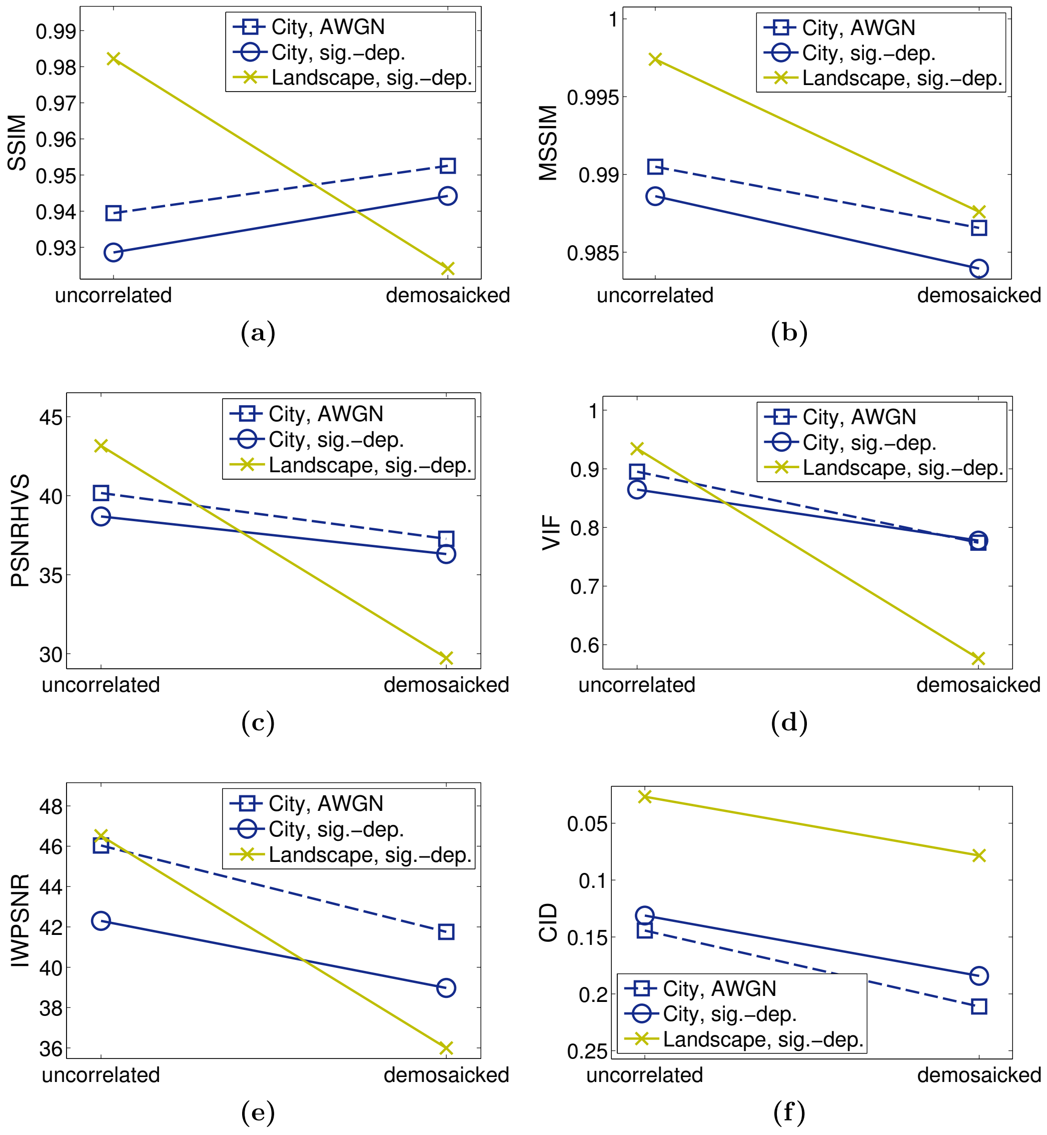
can change the difference and the relative order of the visual quality of the test sequences.

The effect of the noise model on the PSNR results, shown in Fig. 5.4b, shows clearly different results than for the MOS. While the decrease with debayering is correct for the landscape sequence, very different results are obtained for the city sequences: the PSNR is approximatly constant when AWGN is compared to AWGN with debayering, and regarding the signal-dependent noise it is about 2dB higher for the demosaicked city image (33.95 dB). With debayering, the quality of the city sequence is rated much higher compared to the landscape sequence, thus the relative order does not correspond to the MOS results.

In this section, we have seen that the noise characteristics have a significant effect on the subjective quality. Interestingly, subjective quality in terms of MOS does not match the PSNR of the test sequences. While the PSNR is still widely used, it is well known that it is not optimal for visual quality estimation. Therefore, we evaluate more sophisticated metrics that can be expected to correlate better with the visual quality in the next section.

Full-Reference quality metrics

To improve the correlation with the perceived visual quality, adaptations of the mentioned PSNR were proposed, such as the visual signal-to-noise ratio (VSNR) [24] and the human visual system based PSNR (PSNR-HVS/ PSNR-HVSM) [35]. The performance of several quality metrics including the above has been evaluated using the TID2008 database in [100] that contains a test setup with different types of noise, including spatially correlated noise. In this test the PSNR-HVS achieved



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.5: Quality metric results for the test sequences "City" and "Landscape" using the traditional AWGN model (dashed) and the realistic signal-dependent noise (solid lines). The uncorrelated noise, processed without debayering, is shown on the left, the results with debayering on the right.

a high correlation with the subjective test results. We therefore can expect good results using this metric for our test data.

Other approaches that showed good results for various degradations, including white noise and Gaussian blur [114], are structural algorithms, as the structural similarity (SSIM) index [124] and the multiscale SSIM (MSSIM) [126], and information-theoretic algorithms, as the visual information fidelity (VIF) [112] and the information fidelity criterion (IFC) [113].

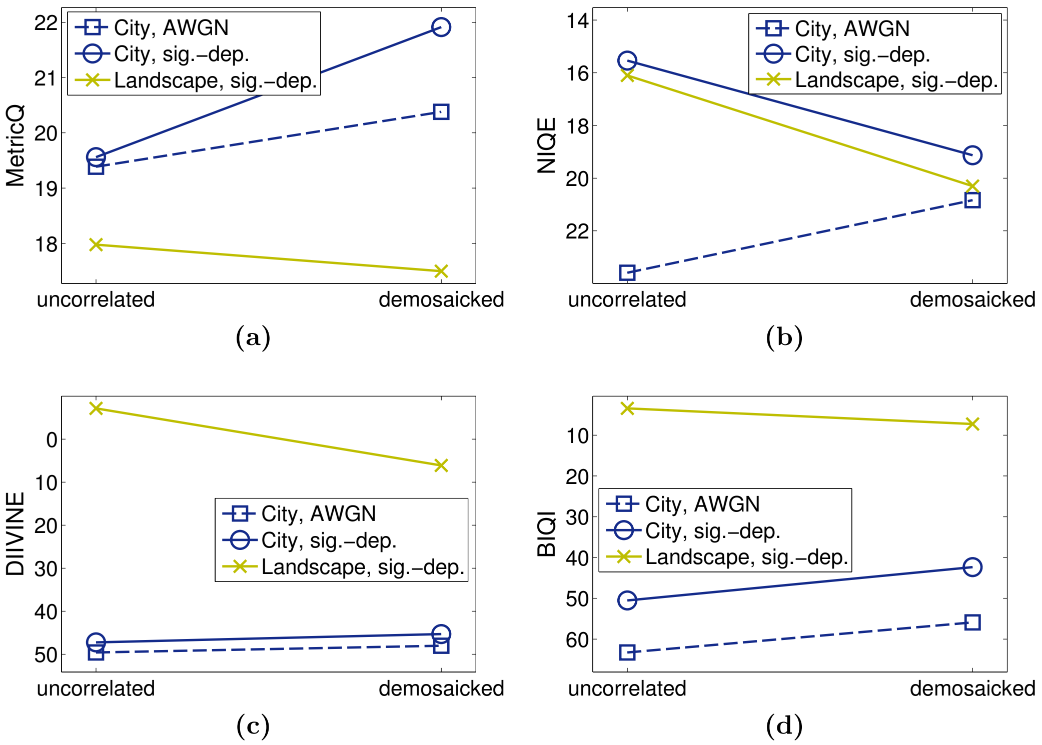
We add three recently proposed metrics that are adaptations of the PSNR and SSIM and showed high correlation based on tests with different standard databases, the PSNR-HMA [99], the "information content weighted PSNR" (IW-PSNR) and the "information content weighted SSIM" (IW-SSIM) [125]. As we work on color image sequences, we add a metric named "color-image-difference" (CID), which is designed for color images specifically [74].

We use the publicly accessible Matlab package "metrix mux" for most full-reference metrics. For PSNR-HVS, PSNR-HMA, IW-PSNR, IW-SSIM and CID we use the Matlab code provided by the authors. While some metrics have parameters that could be tuned, we stick to the default parameters for all the test sequences to achieve comparable results.

An optimal quality metric would show the same tendencies and relative order as the MOS results described in the previous section. One of our main results from the previous section was that including the debayering step in the processing pipeline of the noisy test images significantly reduces visual quality. That means, for both noise models the MOS is lower when debayering is included. In contrast, the SSIM is higher for the spatially correlated noise with debayering (Fig. 5.5a), the relative order of the sequences including debayering is thus incorrect.

The MSSIM better matches the subjective quality, as it shows a lower value for the noise after debayering, see the MSSIM in Fig. 5.5b. IW-SSIM is not shown, as the results are very similar. Also the VIF, IFC, PSNR-HVS, PSNSR-HMA and IW-PSNR and CID rate the noisy images including debayering lower and thus reflect well the subjective quality, Fig. 5.5c shows PSNR-HVS and 5.5d the VIF. CID in Fig. 5.5f was included as it specifically works on color images. Despite that the noise seems very colored in the demosaicked landscape sequence, the quality of is rated very high using CID, which is incorrect. Additionally the order of AWGN and signal-dependent noise does not match the MOS result.

In summary, none of the tested metric is able to perfectly reflect the visual quality of all the noisy sequences as given by the MOS. However, most of the metrics showed lower results for the noisy images when debayering is included compared to the uncorrelated noise, and thus matched this important tendency shown by the MOS.



(a) (b) (c) (d)

Figure 5.6: No-reference quality metric results for the test sequences “City” and “Landscape” using the traditional AWGN model (dashed) and the realistic signal-dependent noise (solid lines). The uncorrelated noise, processed without debayering, is shown on the left, the results with debayering on the right.

No-reference quality metrics

A test setup to compare denoising algorithms usually works with pairs of a reference and the corresponding degraded noisy image. However, in real applications that require a denoising step, no reference image is available, which is the reason why real camera data cannot be used directly for a quantitative evaluation. To make use of real-world data that does not provide a reference, e.g., the new image test set proposed in [11], metrics that can evaluate the visual quality without a reference would be necessary. Besides, automatic parameter tuning would require such NR metrics. The best denoising parameters depend on the image content, and NR metrics would enable to adjust the parameters for each image adaptively. We thus extend our recent work [10] to general no-reference (NR) metrics evaluation.

The recent progress in NR metrics is encouraging. In [132] a metric for parameter tuning, the “MetricQ” is shown to enhance denoising performance. In 2010, Moorthy and Bovik published the NR metric “blind image quality index” (BIQI) [91]. The metrics are usually evaluated using the LIVE database [114] that provides five different types of degradation: JPEG, JPEG2000 (JP2K), white noise, blur and fast fading. The two most relevant distortions for denoising are white noise and blur; we thus compare and select the methods mainly based on the Spearman correlation for these two types of distortion. For both distortion types a higher correlation than PSNR is reported for BIQI, so we include it in our test.

Since then, five other quality metrics have been published. The "general regression neural network (GRNN)", proposed 2011 [70], can provide results highly correlating to the human perception when the right dataset was chosen in the training. For in the most relevant criteria, white noise and blur, the correlation results strongly depends on the dataset: a high correlation for WN comes with a low correlation for blur and vice versa. Therefore we conclude that it may not be appropriate for denoising tests.

The NR metric "distortion identification-based image verity and integrity (DIIVINE)" [90] shows a very high correlation with the mean opinion score of the LIVE database of 0.98 for the white noise distortion and a considerable correlation coefficient of 0.92 for blur. It thus clearly outperforms a NR metric named BLIINDS [106]. An upgraded version thereof, named BLIINDS II [105], was proposed in 2012. However, we do not use it in our test, as it shows an extremely low correlation for white noise (0.1) and thus does not seem to be appropriate for denoising tests.

Two other recently published NR metrics show promising results: the "blind/referenceless image spatial quality evaluator" (BRISQUE) [87] shows a lower correlation for white noise compared to DIIVINE, however, the correlation coefficient is higher for blur. We thus include it in our tests. NIQE [88], which stands for "natural image quality evaluator", outperforms DIIVINE and BRISQUE in both white noise and blur and thus is the most promising method included in our test.

Fig. 5.6 shows the NR-metric results for the noisy test data. MetricQ shows a very high quality for the demosaicked sequences compared to the sequences with uncorrelated noise, whereas the MOS indicates a lower quality for the demosaicked sequences. The same problem we have seen before with the FR metrics SSIM and PSNR. We tried to calculate the anisotropy map differently, however all the MetricQ versions were not able to reflect the decrase when debayering is included for the noisy data. As the results of the difference versions are very similar, we show only the original version in Fig. 5.6a.

While BIQI shows the same problem (Fig. 5.6d), the metric NIQE in Fig. 5.6b shows for a part of the sequences a lower quality when debayering is included, it matches the MOS results for the sequences with signal-dependent noise. While DIIVINE in Fig. 5.6c shows the correct tendency for the landscape sequence, it rates the demosaicked sequences with higher quality for the city sequences, which does not match the MOS results. Regarding the relative order of the test sequences, the MOS value only shows a small difference in the absolute quality of the two sequences, while, according to BIQI and DIIVINE, the landscape sequence has a clearly higher quality rating.

While most metrics, except CID, rated the sequences containing signal-dependent noise lower than the sequences with AWGN, all no-reference metrics rate the sequences with signal-dependent noise higher compared to the AWGN sequences. None of the tested metrics, however, evaluated the sequences corresponding to the MOS, which indicates higher quality of the sequence with signal-dependent noise compared to the AWGN sequence when debayering is included, but a clearly lower quality for the sequence with uncorrelated signal-dependent noise compared to the sequence with uncorrelated AWGN.

The NR metrics seem to less reflect the MOS result than the FR metrics. Especially the relative order of the sequences with different image content is more difficult to estimate without a reference.

5.1.3 Quality Assessment of Denoising Results

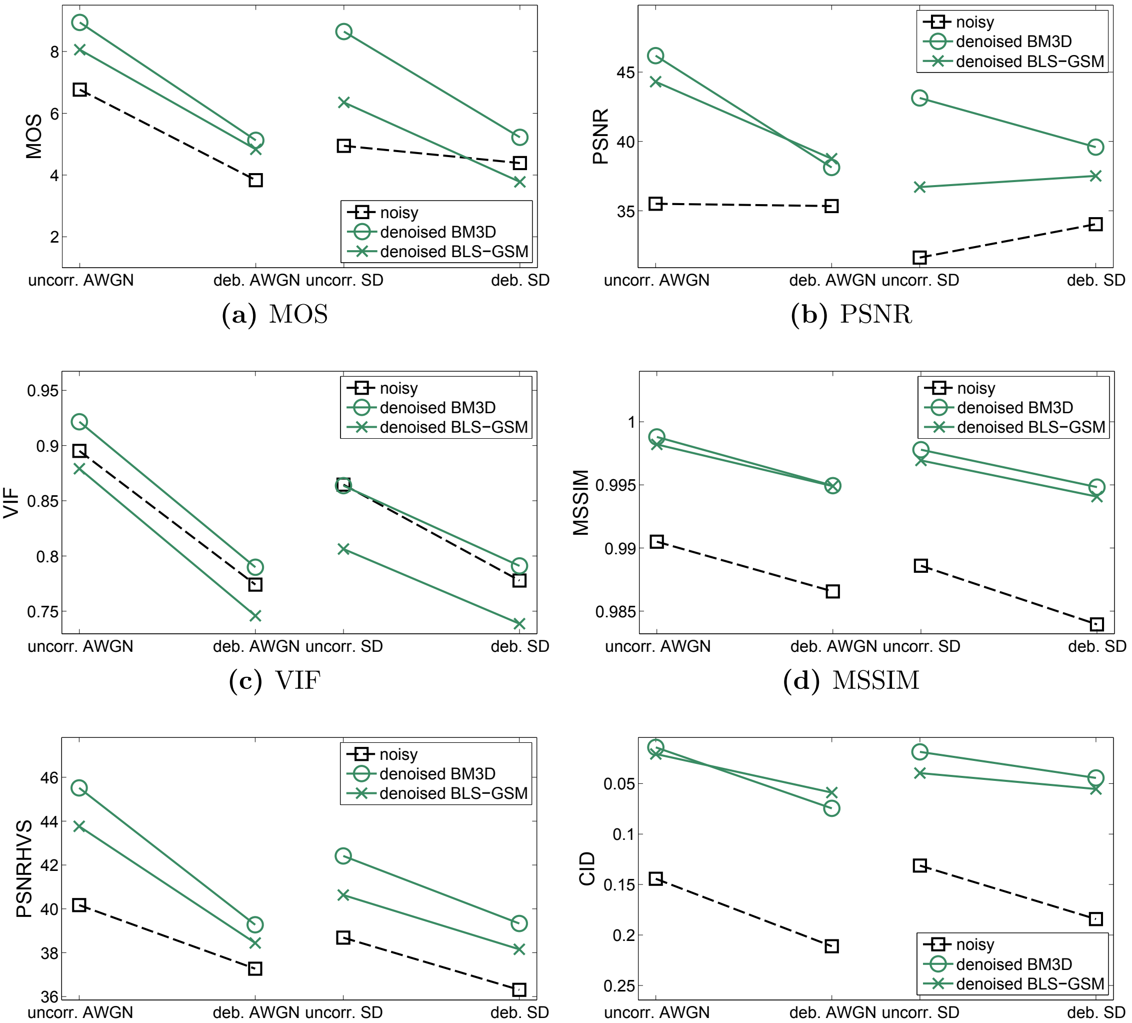
In the last section we discussed the subjective quality and the quality metric performance for noisy sequences. In this section, we discuss the visual quality of the denoised test sequences and based on these results we evaluate the performance of the quality metrics for the denoised test images. Finally we identify the most suitable quality metrics for denoising algorithm evaluation.

Subjective quality

We used two different state-of-the-art denoising methods, BM3D [33] and BLS-GSM [101]. In Fig. 5.7a the MOS results for denoised sequences are shown. In the subsequent plots the two left results show the quality of the usual AWGN model and the AWGN model with debayering, whereas the two results on the right show the quality for signal-dependent noise and for signal-dependent noise with debayering. The last corresponds to the realistic camera noise model.

The noise model significantly affects the visual quality of denoising results. Including the debayering step into the noise simulation, leads to a MOS that is up to 3 scores lower. Thus, denoising uncorrelated noise is easier than denoising spatially correlated noise. The signal-dependence also affects the denoising performance: BLS-GSM achieves a 1.7 lower MOS for uncorrelated signal-dependent noise compared to AWGN. When both signal-dependence and debayering is included, thus realistic camera noise is used, the quality of the BLS-GSM result is even lower than the quality of the noisy sequence, thus no improvement is achieved with denoising. This shows that it is crucial to use an adequate model in the development of denoising algorithms.

We found that the noise model has a significant effect on the visual quality of denoising results. We have shown that debayering decreases denoising performance



(a) MOS

(b) PSNR

(c) VIF

(d) MSSIM

(e) PSNR-HVS

(f) CID

Figure 5.7: MOS and full-reference quality metric results for the denoised test sequences using BM3D [33] and BLS-GSM [101].

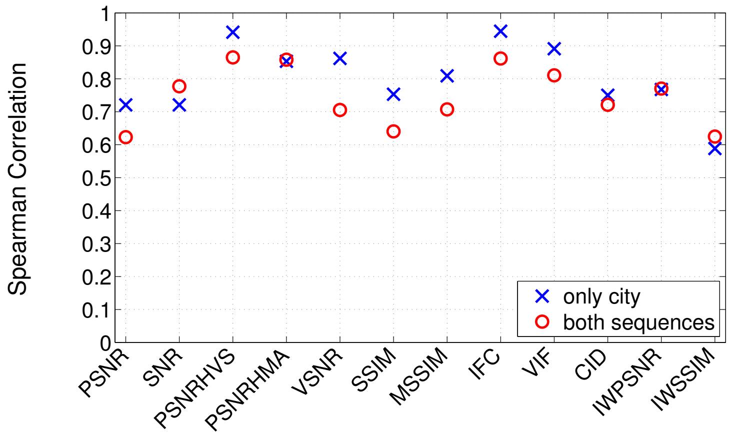


Figure 5.8: Spearman correlation coefficients of full-reference metrics for the visual test.

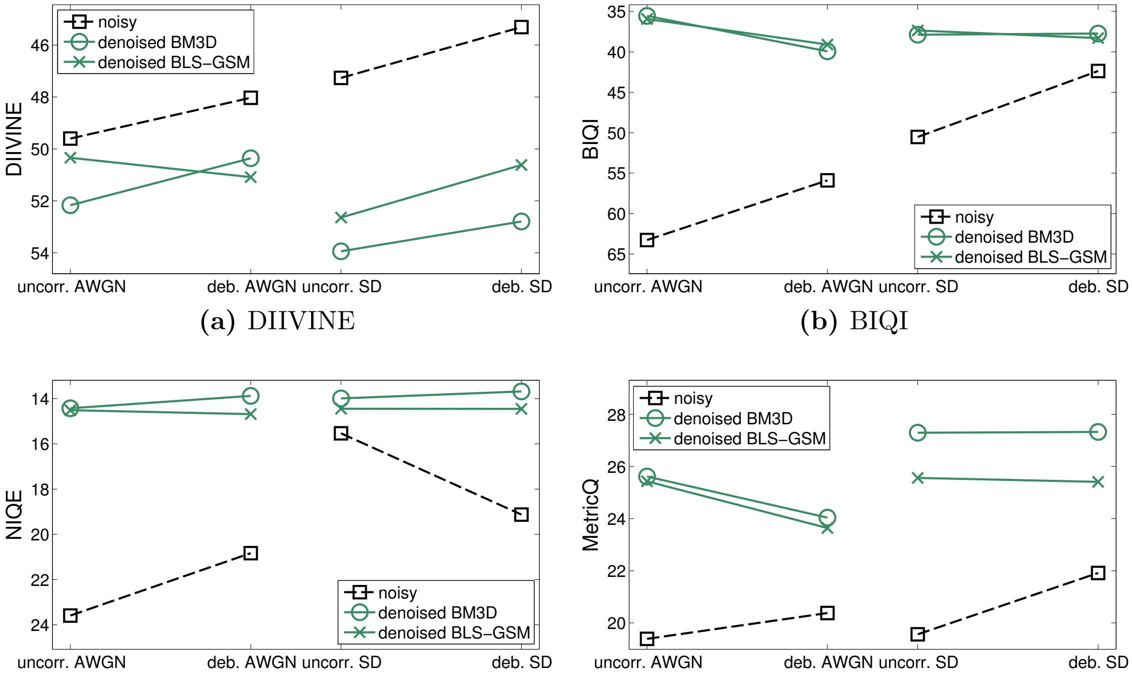
of both tested algorithms significantly and found that the signal-dependence leads to a lower visual quality of the BLS-GSM results.

Full-Reference quality metrics

Optimal quality metrics should show the same tendencies and relative order as the MOS results. As described above, the debayering leads to a lower MOS, thus the lines in Fig. 5.7a are decreasing. This is reflected by the MSSIM, VIF, PSNR-HVS and CID, see Fig. 5.7c-5.7f. The PSNR in Fig. 5.7b matches this tendency for the denoised sequences except for the sequence with signal-dependent noise denoised with BLS-GSM. The PSNR-HMA shows very similar tendency like the PSNR-HVS and thus is not shown. Regarding the relative order of the sequences with realistic camera noise, the VIF is the only metric in our test that rates the denoised sequences using BLS-GSM lower than the noisy sequence, which reflects the MOS (see "dem. SD" in Fig. 5.7c). However, the VIF rates the sequences denoised with BLS-GSM lower for the other noise types, too, which does not reflect the MOS results.

Comparing our subjective test results to the quality metric results shows that none of the tested metrics reflects the MOS perfectly. To determine the metrics that correlate best with the perceived quality, we evaluate the overall metric performance by means of the Spearman correlation coefficient.

Fig. 5.8 shows the correlation coefficients for the full-reference metrics. The correlation coefficients with and without the Landscape sequences are shown. Most of the metrics show lower correlation coefficients when both test sequences are compared. The most stable results are achieved using PSNR-HMA and IWPSNR. The highest correlation to the subjective quality over the entire test is achieved by the full-reference metrics IFC, VIF, PSNR-HVS and PSNR-HMA. Thus, among the tested metrics these are the most suitable for the evaluation of denoising algorithms.



(a) DIIVINE **(b)** BIQI

(c) NIQE **(d)** MetricQ

Figure 5.9: No-reference quality metric results for the denoised test sequences using BM3D [33] and BLS-GSM [101].

No-reference quality metrics

The visual quality of the denoised sequences is higher than the noisy sequences, except the BLS-GSM algorithm applied to realistic camera noise. BM3D shows a higher MOS rating than BLS-GSM and debayering leads to a lower visual quality of the denoising results. An optimal metric would reflect these findings.

While the MetricQ indicates a lower quality for the demosaicked and denoised AWGN test sequences in Fig. 5.9d, it does not show this tendency for the signal-dependent noise with debayering. We tested three versions in addition to the original version: MetricQ with an anisotropy map that was calculated on the same noisy images (MetricQ AM noisy). This is a reasonable setup, as the noisy image is known when denoising is applied. Additionally, we included two versions that are using the reference data partially. In a second new version, the MetricQ with the same anisotropy map (MetricQ AM const noisy) is used for all the different degradation types. As the authors proposed to use the most degraded image in this case, we used the noisy images with debayering included. In the third version the anisotropy map was calculated on the reference image (MetricQ AM const ref). Fig. 5.10 shows the Spearman correlation coefficient over all the test sequences for the different variants of MetricQ. Compared to the original MetricQ, the variants using degraded

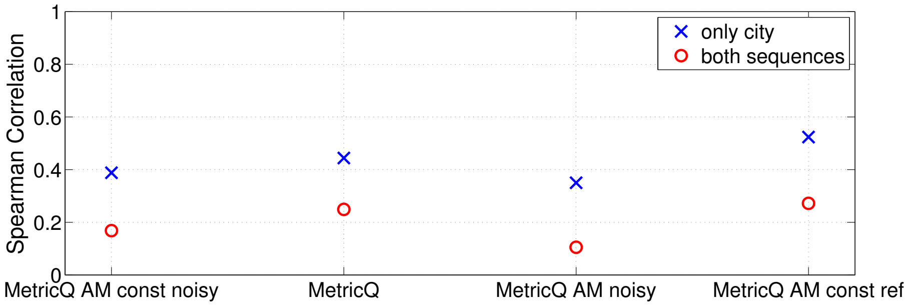


Figure 5.10: Spearman correlation coefficients of MetricQ variants

images correlate less with human perception. The MetricQ with the anisotropy map calculated on the reference image (MetricQ AM const ref) shows a higher correlation.

While DIIVINE rates the denoising results with debayering higher compared to the denoised sequences without debayering, BIQI, NIQE and MetricQ show a constant or slightly decrease in quality when debayering is included before denoising, thus partially match the MOS result. DIIVINE rates all the noisy sequences higher than the denoised sequences. This does not match the MOS result. This may be due to a high sensitivity to blurring in the DIIVINE metric.

The noise model has a significant effect on the visual quality of denoising results and none of the NR metrics in our test fully reflects the MOS. To determine the NR metrics that correlate best with the perceived quality, we evaluate the overall metric performance by means of the Spearman correlation coefficient, shown in Fig 5.11. The NR metrics achieve a much lower correlation compared to the FR metrics. The highest correlation coefficient is 0.6 for the BIQI metric, which additionally seems to be most robust to the image content. While NIQE and MetricQ achieve a considerable correlation coefficient of 0.4 on the city test set, BRISQUE practically does not correlate to the MOS results (0.08) and DIIVINE even shows a negative correlation coefficient, as it rates all the denoised sequences with a lower quality than the noisy sequence. We thus did not find a no-reference metric reflecting the MOS in our test.

While the correlation coefficients given in the literature usually are around 0.9, we have seen relatively low correlation coefficients between visual quality and metric result. This is mainly due to the different approach of calculating the correlation coefficients: in the literature correlation coefficients are usually given for a single degradation type (e.g., white noise) separately. This means the correlation coefficient represents how well a metric can compare a noisy image to another noisy image. In denoising evaluation, however, the metric must cope with different degradations that occur in one image simultaneously and find the best tradeoff, i.e., the best image. Thus the correlation coefficient has to be calculated over the denoised and the noisy test sequences to obtain useful information about how well a metric is suited for denoising evaluation.

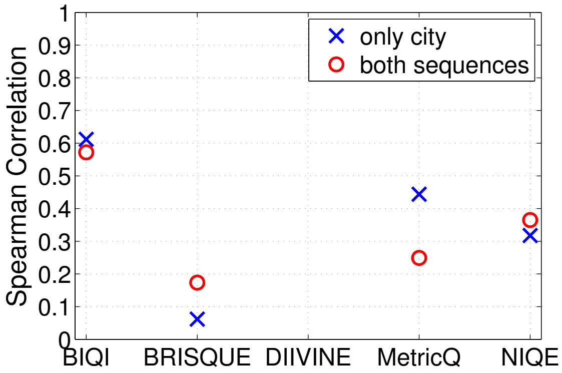


Figure 5.11: Spearman correlation coefficients of no-reference metrics for the visual test. DIIVINE shows a negative correlation, which is probably due to the low quality rating for the denoised sequences.

We mentioned two reasons to use NR metrics: first, these metrics would enable us to directly use test data without reference, and thus real camera data could be used for denoising evaluation. However, as the correlation to human perception is much higher using FR metrics, we conclude that future denoising tests should be based on test sets including reference data. The second reason was that parameter tuning could be done automatically using a NR metric. Our test results indicate that BIQI could be a possible candidate for a parameter tuning metric. However, we need to verify this in future tests including more different sequences and different denoising results obtained by varying parameters.

5.1.4 Conclusion

In this section, we proposed a realistic camera noise model and showed how to integrate the complete camera processing chain into the test setup to evaluate denoising approaches. We not only showed that the noise characteristic of the typically applied AWGN is fundamentally different from our realistic noise model, but also identified that the signal-dependent noise as well as its spatial correlation has a significant impact on the perceived visual quality of noisy and denoised images. In our subjective test, we found that the spatially correlated noise, introduced in the debayering step, is perceived as more disturbing.

Denoising test images degraded by signal-dependent noise leads to results with a lower visual quality than for AWGN. Further, the performance of state-of-the-art denoising algorithms is considerably impaired by spatially correlated noise. Using realistic camera noise, denoising can even reduce the perceived visual quality.

To estimate the visual quality of denoising results without costly subjective tests, reliable quality metrics are required. In this section, several state-of-the-art quality metrics are evaluated. No-reference metrics as well as full-reference metrics have been tested, however, none of the tested metrics fully reflects the perceived visual quality. While the widely used full-reference metrics PSNR and SSIM show a low correlation with the subjective test results, the highest correlation is obtained using the metrics PSNR-HVS, IFC, VIF and PSNR-HMA. All the tested no-reference metrics show a very low correlation to the test results.

With the significant impact of the chosen noise characteristic on the visual quality of noisy images and of denoising results, we conclude that a realistic noise model should be used in future research. For realistic denoising evaluation, a new extensive test set based on realistic noise, as well as new quality metrics that better reflect the visual quality of camera data, would be required.

5.2 Noise perception

5.2.1 Visibility of spatio-temporal noise: introduction

Noise is an unavoidable effect in real-world video sequences, seriously reducing the visual quality of video, especially in low light conditions. With today's high resolution sensors having a smaller pixel pitch, the need for algorithmic solutions is heightend.

In this work, we study the visibility of noise. Our motivation originates from observations in motion picture denoising. An extensive set of denoising methods has been developed to reduce the noise in image and video data after capture. Of course none can perfectly reconstruct the true image; an estimation error always remains. The challenge is to hide this estimation error as well as possible. We observed that this estimation error is especially severe when a spatial image denoising method is applied to video data. The estimation error, or remaining noise, in this case is usually very low frequency noise. This low frequency noise is hidden in still images, because it is almost invisible due to the decrease in the contrast sensitivity of human vision for very low spatial frequencies.

If the images are part of a video sequence, this low frequency noise reappears as flickering in the video sequences and thus disturbs the viewing experience. As the noise in the low frequency bands is very difficult to separate from real image content, it is still a common quality issue in video denoising.

Although temporal denoising methods can better reduce flickering, they may also introduce new artifacts, especially motion artifacts, and they come with high computational cost; memory requirements in particular are extremely high for the current high resolution data (4k and beyond).

To the best of our knowledge no study is available that provides details about the visibility of noise in video sequences depending on their spectral distribution. Winkler and Süsstrunk [145] presented a subjective study examining noise visibility in still images. In addition to white noise, they also used mid-frequency and high-frequency noise. The results show lower noise visibility for the high-frequency noise compared to the mid-frequency noise, as would be expected from the contrast sensitivity function. We expect that the same effect could have been shown for low frequency noise, as contrast sensitivity also falls off towards low frequencies. Unfortunately this has not been a subject of the study.

In this section, we study the visibility of noise for eight different frequency bands. We evaluate the visibility of both static noise, as it occurs in still images, and dynamic noise, as it is present in video data. We start by giving a short overview of the literature about spatial and temporal contrast sensitivity of human vision in Sec. 5.2.2. We then explain how we obtain the different noise patterns, which can be displayed on a standard monitor (Sec. 5.2.3). The results of a test with nine observers for static noise and a dynamic (spatiotemporal) noise are presented in Sec. 5.2.4. We discuss and compare our results in Sec. 5.2.5, and conclude in Sec. 5.2.6.

5.2.2 Related Work

Technical developments like movies and discharge lamps provoked early experiments on temporal effects of human vision. Contrast sensitivity of time varying signaly and how effects can be explained using Fourier analysis was already studied almost sixty years ago by de Lange ([154–158]), Kelly ([147,148]), and Roufs ([139–143]). An overview of the early experiments is given by Kelly in 1977 [152].

The first measurements for combined spatiotemporal sensitivity were presented by Robson in 1966 [159]. He determined the thresholds for four spatial frequencies and four temporal frequencies. The lowest frequency was 1 Hz, which is considered to be equivalent to static results (Van Nes result's indicate threshold modulation a little bit above at 0 Hz [160]). The results indicate that while the spatial CSF measured for static patterns shows a band-pass characteristic, the sensitivity function for the same spatial frequencies, measured with spatial patterns that are temporally varying, becomes a low pass. The spatiotemporal contrast sensitivity is thus not separable, it shows a clearly more complicated shape than could be obtained by the product of spatial and temporal CSFs. Kelly in 1966 already mentioned effects not explainable by a separable model [149] and measured the CSF for spatiotemporal stimuli (travelling waves) [150]. The results are similar to Robson's: for 2 Hz the band-pass shape holds, the frequencies 13.5 Hz, 17 Hz and 23 Hz suggest a low pass structure. For the chromatic CSF curves, refer to Kelly's later measurements [151,153]. Most applications that use spatiotemporal CSF models still rely on this data, e.g., the perceptual quality metric by Winkler [146].

The experiments, however, were all conducted with sine waves. While this has become the standard procedure to measure CSF curves, this section evaluates the sensitivity to noise of different spatial frequencies, as it can occur in video sequences.

Some studies examined the visibility of signals when noise is added [161–163]. However, for the application of video quality assessment and denoising evaluation it is more important to study the visibility of the noise.

5.2.3 Noise visibility Test

We investigate the visibility of noise of different spatial frequency bands in still images and in video sequences (we use 24Hz video sequences). To that end, we conduct a subjective test. We include two types of test patterns in the test: static (spatial) noise patterns and dynamic (spatiotemporal) noise. Eight spatial frequency bands were used for each pattern type.

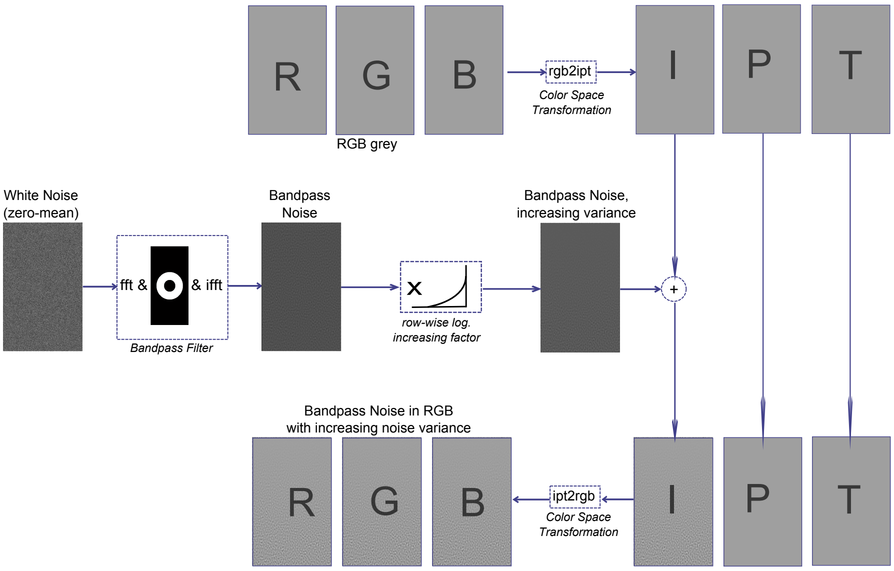

R
G
B
RGB grey
rgb2ipt
Color Space Transformation
I
P
T
White Noise (zero-mean)
fft &
& ifft
Bandpass Filter
Bandpass Noise
x
row-wise log. increasing factor
Bandpass Noise, increasing variance
+
Bandpass Noise in RGB with increasing noise variance
R
G
B
ipt2rgb
Color Space Transformation
I
P
T

Figure 5.12: Workflow for generating the test stimuli for the noise sensitivity experiments.

Test Pattern Generation

The test patterns were obtained by first generating white noise and subsequent band-pass filtering by cutting the desired frequency band in the Fourier domain. To separately test for the sensitivity of lightness and chroma channels, we used the IPT color space and added the noise in only one of the three channels. IPT is an opponent color space that was developed by Ebner and Fairchild [144] to create a space that is perceptually uniform but still easy to calculate.

The workflow for the noise pattern generation is shown in Fig. 5.12 and described in the following.

1. First, 2-D zero-mean white noise is generated by Matlab's randn-function

2. The noise is transformed to the Fourier domain and a band-pass filter is used to cut a defined band from the spectrum.

3. Each pixel row of the image is multiplied by a factor that increases logarithmically from top to bottom, to obtain noise with increasing variance (i.e., contrast).

4. A uniform gray image, with a constant pixel value of 0.6104, is generated in sRGB color space.

5. The gray image is transformed to the IPT color space.

6. The zero-mean noise is finally added to one of the channels of the IPT image.
7. The noisy image is transformed back to sRGB.

The test patterns for the dynamic noise measurements are the same as those for the static noise. We generated 50 static noise patterns, which are displayed as a video sequence with 24 frames per second for the dynamic noise experiments.

Test Setup

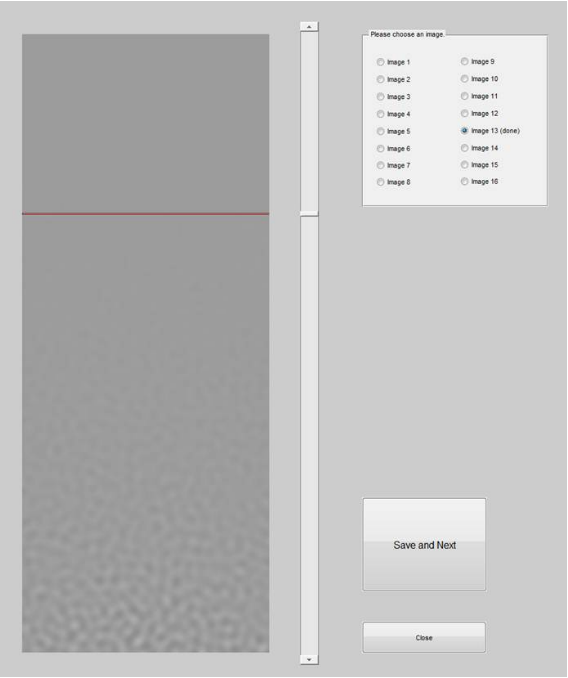


Figure 5.13: Test setup: the red line was adjusted by the observers to match the visibility threshold.

The stimuli are a rectangular field of the test patterns with increasing contrast from top to bottom, Fig 5.13 shows a screenshot. The concept of the test is to directly measure the sensitivity by letting the observers choose the limit of visibility.

On the left, the test pattern is displayed (static noise in the achromatic channel in this example). The subject is asked to move the red bar, by using the slider right to the test pattern, to the highest possible point where the test pattern is still visible. With the radio buttons on the right, different frequency bands can be selected. As eight frequency bands are available the subject has to evaluate eight images. When clicking on “Save and Next”, the slider positions for all images are saved and the GUI for the temporal measurements is loaded. The next test is dynamic, achromatic noise. When the observer has completed the adjustments for all the achromatic patterns, the GUIs for the chromatic patterns are loaded, first static and then dynamic measurements.

The test patterns were displayed on an sRGB monitor with a viewing distance of 150 cm. We did the measurements for 8 frequency bands between 0.14 and 27.62 cpd. Tab. 5.1 shows the frequency bands of the noise test patterns.

Table 5.1: The frequency bands used in the test are given in pixels per period and cycles per degree.

Nr. of pixels per period		Spatial frequency in cycles per degree	
range	mean	range	mean
311.11 - 700	505.55	0.14 - 0.31	0.19
155.56 - 311.11	233.34	0.31 - 0.62	0.41
77.78 - 155.56	116.6	0.62 - 1.24	0.83
38.89 - 77.78	58.335	1.24 - 2.48	1.66
18.67 - 38.89	28.78	18.67 - 38.89	3.36
9.33 - 18.67	14	5.18 - 10.36	6.9
4.67 - 9.33	7	10.36- 20.70	13.81
2.33 - 4.67	3.5	20.7 - 27.62	24.16

5.2.4 Results

Based on the row that the nine observes chose, we calculate the contrast sensitivity, Fig. 5.14 and Fig. 5.15. We additionally plotted bars indicating the standard deviation of the different votes. The bars show that the votes deviate from observer to observer significantly. Hence, it may be worth extending the excellent and extensive early research, e.g., by Kelly, beyond the experiments based on a single observer.

The static noise sensitivity in Fig. 5.14 shows a peak in the I-channel curve around 1.66/3.36 cpd. The chromatic curves, however, show a peak shifted to lower frequencies, the maximum sensitivity is in the three lowest frequency bands (0.19, 0.41 and 0.83 cpd). This finding matches quite well the shape of the curves measured by Kelly for unstabilized stationary sine gratings in [153]. One difference to the previously reported curves is the relation between achromatic and chromatic channels. As in previous results [151] the achromatic gratings induce higher sensitivity than chromatic gratings, one could have expected that the P- and T-channel noise would also show lower sensitivity than the I channel. However, this is not the case: the peak sensitivity for the I-channel noise is not higher than for the P- and T-channel noise – it is only shifted to lower frequencies. The reason for that can be explained by the design of the IPT color space. It is designed to be perceptually uniform, and therefore an I-channel contrast should correspond to the same perceived contrast in the P- or T-channel.

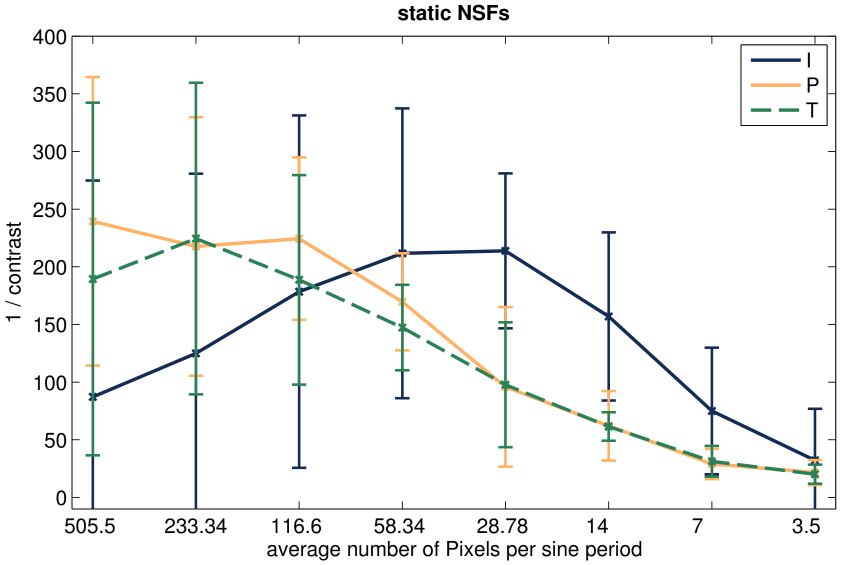


Figure 5.14: Test with the static noise patterns: mean results are averaged over all observers adn plotted with the standard deviation of the results.

The dynamic noise results are very different. Fig. 5.15 shows that the sensitivity in the achromatic I-channel is clearly higher than in the chromatic channels. Further, the peak is completely shifted to the lowest measured frequency for all three sensitivity curves.

This explains what we observed in videos that were spatially denoised: whereas the denoising result, viewed as a still image, looks unimpaired, a video sequence composed of these images shows disturbing flickering.

5.2.5 Discussion

Comparison of test results to the contrast sensitivity function

Due to the clear decrease of the temporal contrast sensitivity above 10 Hz, one would have expected that a flickering grating displayed at 24 Hz could lead to lower sensitivity than the static spatial noise. However, our results show the contrary. In Fig. 5.16 we replotted our previously shown results in the XYZ color space, instead of IPT, to make them more comparable to other publications.

In the low and mid-frequency range, sensitivity is clearly higher for dynamic noise compared to static noise. This matches the observed problem we have seen in the visual quality of denoised videos in contrast to the good denoising results achieved by applying the same algorithms to still images. In the high frequency bands around 13.8 cpd and 24 cpd, sensitivity to the dynamic noise is lower than to the spatial noise, but the difference is not large.

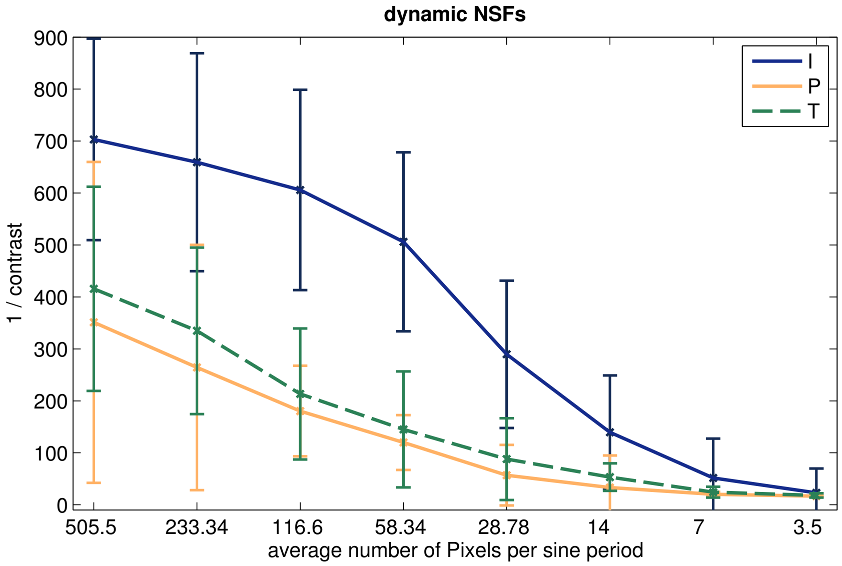


Figure 5.15: Result of the test with the dynamic noise, which is a sequence of newly generated static noise patterns with 24 frames per second: mean results are averaged over all observers adn plotted with the standard deviation of the results.

The absolute value of the contrast sensitivity is hardly comparable, as it depends on parameters like mean luminance and the presentation of the patterns, e.g., how many cycles are shown. Campell et al. found that the contrast sensitivity for a single sine wave is lower than for the standard repetitive sine wave grating [164]. However, a fixed image size forces the cycle number to vary depending on the frequency. Future experiments should be performed with different image sizes to study the influence of the number of cycles.

As we used a standard monitor, the quantization with 8bit can unintentionally affect the shape of the signal at very low contrast levels. It can lead to a grating that would be more comparable to a rectangular pattern than a sine wave. This would explain higher higher sensitivity values in our test, since a square wave grating has been shown to lead to higher contrast sensitivity than a sine wave grating in the low frequency range by Campbell et al. in [165]. In future experiments, a 10bit monitor would be preferable to be sure that the quantization is beyond perceptional limits. However, as an 8bit monitor is the usual display, our results are relevant for this very realistic use case.

The absolute values of the contrast sensitivity function are higher than reported by Robson [159] and Kelly [150]. Besides the above described effect of 8bit quantization, the test setup can be a reason: the users drew the line at a point where they were sure not to perceive any noise. This leads to a higher sensitivity than the one in the

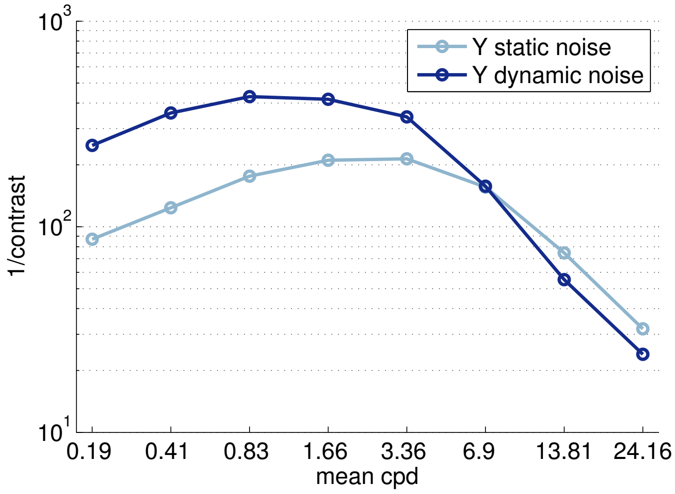


Figure 5.16: The contrast sensitivity was calculated in XYZ color space for better comparison with other results. Plotted here, is the luminance(Y-channel) results of the static noise patterns and the dynamic noise patterns.

usual tests, in which observers adjusted the modulation such that the grating is just visible. Additionally, the luminance levels of the screen do not match exactly and the surrounding luminance may also be different. Whereas the above mentioned reason explain that absolute sensitivities do not match previous results exactly, our goal in this work was not to reproduce the extensive research that has been done in the last decades. Our goal was to investigate the visibility of spatial noise in still images and compare it to the spatiotemporal noise in video sequences. For this relative comparison, our measurements provide meaningful data.

The main result of comparing the visibility of spatial static noise to spatiotemporal dynamic noise is that the noise of spatially low frequencies is clearly more visible in the dynamic noise test than in the static noise test. This result matches the clear visual impression and should be considered for improving research on denoising for video sequences.

Perceived lightness vs. luminance measurements

When comparing the results in IPT (Fig. 5.14 and 5.15) to the curves that show the contrast in terms of luminance (Fig. 5.16), the contrast sensitivity function varies in shape. This is due to the nonlinear mapping of luminance and perceived lightness, which is also implemented in other perceptual color spaces, e.g., Lab. Although it is practical to use XYZ for the comparison with the available literature, we have the impression that the shape in the IPT contrast sensitivity matches better our visual perception because the dynamic noise seems to increase towards lower spatial frequencies, which is well represented by the I-channel results, shown in Fig. 5.15.

Sensitivity increase

Tab. 5.2 shows the sensitivity compensation factors (S) from the static noise contrast sensitivity Y_{stat} to the dynamic noise contrast sensitivity Y_{dyn}.

$$S = \frac{Y_{dyn}}{Y_{stat}} \tag{5.1}$$

These factors can easily be included in metrics that already include the CSF to adapt them to video. This can be expected to improve the quality estimation of noisy and spatially denoised videos.

Table 5.2: Sensitivity compensation factors calculated for the luminance contrast sensitivity in Fig. 5.16.

mean spatial frequency	sensitivity compensation factor
0.19 cpd	2.9
0.41 cpd	2.9
0.83 cpd	2.4
1.66 cpd	2.0
3.36 cpd	1.6
6.9 cpd	1.0
13.81 cpd	0.7
24.16 cpd	0.8

5.2.6 Conclusion

We measured the visibility of noise for eight spatial frequency bands. The subjective test included both static noise, as it can occur in denoised still images, and dynamic (spatiotemporal) noise, as it can occur in denoised video data. Comparing the dynamic results to the static results with the same spatial frequency yields two main results.

First, the contrast sensitivity of spatially low frequency noise is significantly higher when the noise is temporally varying. Due to the decrease of the temporal sensitivity with higher temporal frequencies, this may be a surprise, but it matches well to our visual impression. This finding can be helpful in denoising video sequences, where the frequency dependence can now be included in the design and optimization of new denoising algorithms. Further, the proposed sensitivity compensation factors can easily be included in metrics that already include the CSF to adapt them to video.

Second, we showed the contrast sensitivity both in terms of perceived lightness (in IPT color space) and for luminance (XYZ color space). The shape in the IPT color space is different – the effect of increasing temporal noise visibility when the noise is low in spatial frequency is even more clear when shown in IPT. We think that this is very interesting, since, for the application in image quality estimation, perceived lightness is more important than the absolute luminance.

The factors proposed to improve existing quality metrics for noisy and denoised video have been obtained from tests with grey background. Future research should therefore focus on using more realistic test stimuli, e.g., real motion picture data, to evaluate how masking of image content influences the visibility of spatiotemporal noise. Secondly, more realistic broader band noise could be included in future tests to evaluate how well our findings can be used for real videos.

6 Realistic Noise Model and Denoising Method for Time-of-Flight data

Whereas the main part of this dissertation aims at improving color image sequences, this chapter presents how data of other cameras can be improved based on gathered experience.

Nowadays, additional information captured together with a color image becomes increasingly important. Combining the color camera with a depth camera creates added value – the 3D information can, e.g., be used to improve interaction between computer graphics and real-world objects, allow for object dependent color grading, or facilitate focus adjustment.

Time-of-flight (TOF) technology enables capturing depth data with relatively high speed and thus can be used to capture depth video data. However, the limitations are the low resolution compared to color imaging sensors, the limited range of distinct depth, and very high noise in the TOF depth images. The resolution may be increased with advancing sensor technology, but the noise can be reduced directly using algorithmic methods. In this chapter, which builds on our previous publication [2], we describe the noise characteristics of the TOF data and present a solution that is specifically designed to reduce this noise.

6.1 Introduction

Fig. 6.1 shows a typical native depth map taken with a TOF camera. Two types of noise can be observed: the first type are fluctuations mainly around the true signal level, usually assumed as Gaussian noise [180, 181]. The second type appears like salt-and-pepper noise, thus noisy pixels appear to be unrelated to the true signal level.

A typical method for denoising TOF data is joint-bilateral filtering (JBF) [174–176, 178]. Whereas the JBF significantly reduces Gaussian noise, the typical solution for salt-and-pepper noise is median filtering [182]. However, median filtering smoothes the image and therefore leads to a loss of details.

We propose to use a new method to remove the salt-and-pepper noise. Our method is not based on the observed noise types but deploys the physical TOF-

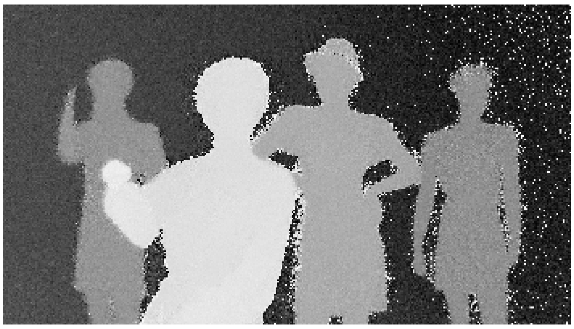


Figure 6.1: Example of a photonic mixer device TOF depth map in grayscale representation (white=0m; black=7.5m; linear in between). Crucial salt-and-pepper Noise is obvious in the right image area

specific causes of the noise. The cause of the salt-and-pepper noise in photonic mixer device (PMD) TOF images is phase ambiguity: due to the cyclic nature of phase, the range of distinct depth values (non-ambiguity-range) is limited. We explain the noise characteristics of the TOF data in detail in Sec 6.3. In the following, we present our solution to the described problem, a novel phase-aware pre-processing filter (Sec. 6.4). The experiments with real TOF data in Sec. 6.5 show the effectiveness of the proposed method, and conclusions are drawn in Sec. 6.6.

6.2 Related work

Typically, salt-and-pepper noise is removed by nonlinear filters like the traditional median filter. Various improved median filtering methods have been proposed to better preserve image details [168, 170, 172]. Median filtering tends to oversmooth the image as it does not distinguish whether a pixel is corrupted. Improvements have been proposed that detect and process only the corrupted pixels [167, 169, 171, 173, 187]. Recently methods for mixed noise removal (AWGN and salt-and-pepper noise) have also been proposed, e.g., by Jiang et al. in [166] and methods mentioned therein. All these methods, however, assume corruption of pixels by effects like faulty memory or bit errors in transmission and thus rely solely on the signal values in a neighborhood for removing the noise. We will show that the corrupted signal in the TOF data still carries valuable information that can be used. The usual assumption behind denoising algorithms is that the correct estimate is the value corresponding to the highest probability density of a signal distribution. Whereas median filtering is more robust to outliers, e.g., the white pixels in Fig. 6.1, it still follows the assumption that many white values indicate a brighter true value and vice versa. However, in the case of TOF noise, a white value can indicate that the true value is very dark, thus close to the end of the non-ambiguity range.

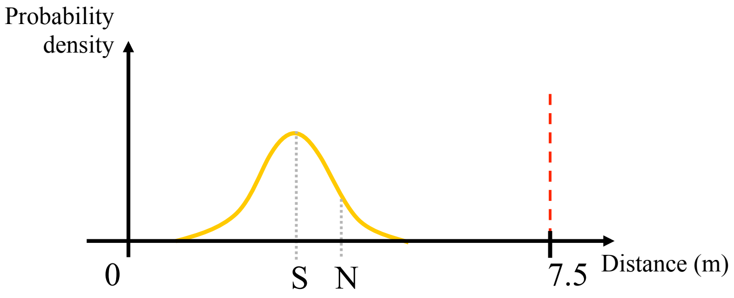


Figure 6.2: Noise characteristic of PMD TOF signals far away from the range limits. PACS does not propose a candidate value because the noisy value N is far enough away from the TOF range limits.

In order to un-wrap phase ambiguity in phase-modulated TOF systems, two different approaches exist.

The first class of methods exploits time-division multiplexing techniques to overcome the modulo principle and hence to extend the measurement range. Therein, at least two modulation frequencies take turns [183,184]. In detail, at least two consecutive TOF images with different modulation frequencies are captured and result in one un-wrapped image out of both, similar to HDR in computational photography. However, this technique is only useful for still images or motion pictures with very slow motion, as it requires static content in both depth images for a robust composite.

The second class of methods estimates the un-wrapped depth using a rough estimate about the true depth based on an empirical amplitude to depth ratio of the TOF signals [185,186]. This principle makes use of the assumption that an object surface located further away reflects less due to the quadratic loss of light intensity. However, different surface reflectance can compensate the quadratic law and to failing of these methods.

As phase unwrapping is thus difficult to be applied in a reliable way for motion picture content, in this work, we are not aiming for range extension but for repositioning the noisy pixels that are wrapped because of the TOF modulo principle. This allows an optimal use of the available TOF range and reduces errors at the borders of the non-ambiguity range. In the following we explain the occurrence of the salt-and-pepper-like noise in detail.

6.3 Range ambiguity noise

As mentioned, our method is based on the physical causes of the noise, thus we first describe the relevant noise characteristics in detail in this section.

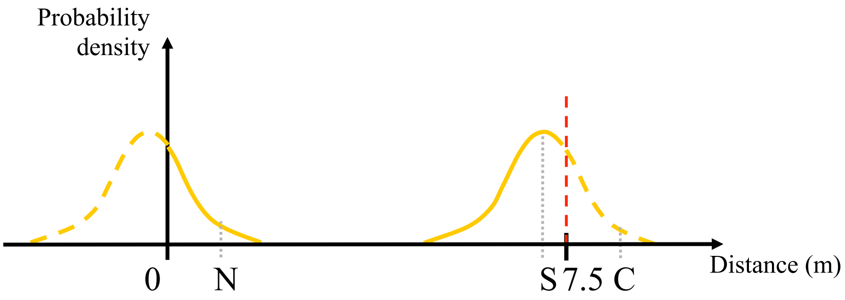


Figure 6.3: The modulo effect of a TOF system illustrated by a signal value S close to the upper signal limit. The measured noisy signal N near the lower limit possibly depends on the Gaussian distribution of the Signal S. PACS proposes C as an alternative candidate value.

The noise characteristic far from the borders of the signal range can be approximated by a Gaussian distribution which is visualized in Fig. 6.2 [180, 181]. In the TOF data, salt-and-pepper noise is not a different noise contribution, instead it is caused by the inherent Gaussian noise. Fig. 6.3 illustrates this issue. When a signal S is located close to the borders of the non-ambiguity range, in our example 7.5m, the Gaussian noise leads to values exceeding the signal range. As the measurement is a phase measurement, this leads to a modulo effect with respect to the non-ambiguity range. That means, value C is mapped to value N near the opposite border of the signal range. Value N is very far away from the true signal S and hence appears as salt-and-pepper noise. For clarification, this noise is called range ambiguity noise (RAN) in the remainder of this dissertation.

Instead of directly estimating the true signal S, we first recover the noisy depth value C in our method. This value is the physically correct depth value affected by Gaussian noise only. This pre-processing trick is crucial, as now the filters originally optimized for Gaussian noise distributions can be used adequately.

6.4 Phase-Aware-Candidate-Selection

We first present our basic method and subsequently describe how to optimally restrict the filter to the affected pixels based on noise measurements.

6.4.1 Method

Our proposed pre-processing filter, the Phase-Aware-Candidate-Selection filter (PACS), estimates the value C based on the spatio-temporal neighborhood $\Omega \times T$. In case that the current noisy depth value N is close to the borders of the TOF range, PACS

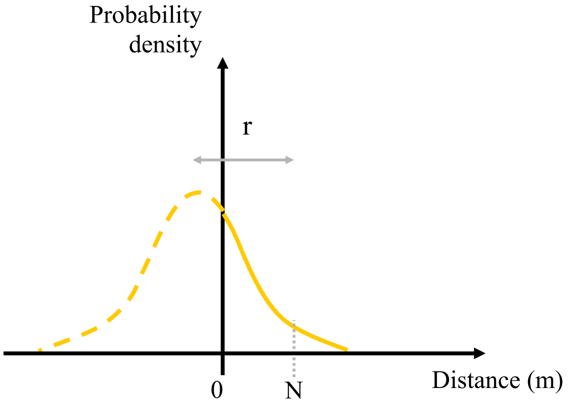


Figure 6.4: The range parameter r is responsible for the application of PACS. In this case filtering is applied.

determines the corresponding candidate value C utilizing the non-ambiguity range d_{nar} (e.g., 7.5m).

$$C = \begin{cases} N - d_{nar}, & \text{if } N > \frac{d_{nar}}{2} \\ N + d_{nar}, & \text{if } N \leq \frac{d_{nar}}{2} \end{cases} \tag{6.1}$$

Subsequently, PACS calculates weighted Euclidean distances d for N and C using

$$d_{N_{p,t}} = \sum_{\tau \in T} \sum_{q \in \Omega} ||N_{p,t} - D_{q,\tau}||_2 \cdot G_\sigma(||I_{p,t} - I_{q,\tau}||_2) \tag{6.2}$$

$$d_{C_{p,t}} = \sum_{\tau \in T} \sum_{q \in \Omega} ||C_{p,t} - D_{q,\tau}||_2 \cdot G_\sigma(||I_{p,t} - I_{q,\tau}||_2) \tag{6.3}$$

G is a Gaussian weighting function.

To take advantage of the nearly noiseless edge information of the color image, the weighting function is based on the color intensities I from the RGB image, which is registered to the depth map. The last step is the decision for the final depth intensity D at the position p:

$$D_p = \begin{cases} C_{p,t}, & \text{if } d_{N_{p,t}} > d_{C_{p,t}} \\ N_{p,t}, & \text{if } d_{N_{p,t}} \leq d_{C_{p,t}} \end{cases} \tag{6.4}$$

6.4.2 σ-LUT thresholding

To enhance the filtering results, we add an additional step that decides if PACS is required. As described in Sec. 2.2, RAN only occurs close to the borders of the non-ambiguity range. Therefore, in this section, we derive a criterion whether a pixel is close to the borders.

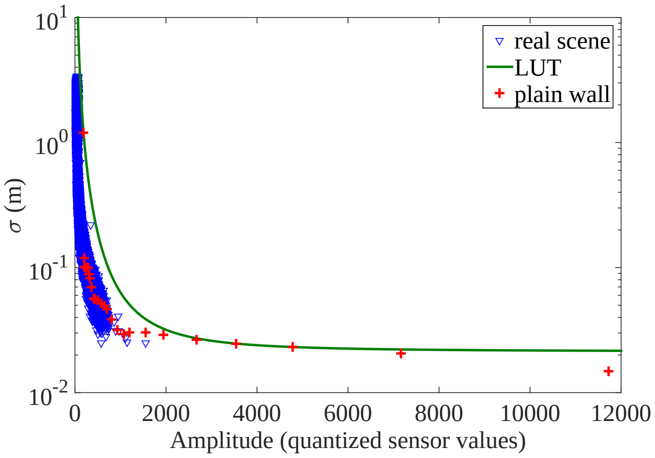


Figure 6.5: The red values are calculated standard deviations of measured TOF data at different distances in front of a plain wall. The green curve is the adapted LUT. The blue values show standard deviations of a realistic static scene with their averaged amplitudes.

According to the Gaussian distribution the probability that a noisy signal is within $S \pm 2\sigma$ is 95.4%. Hence, a deviation between the noisy signal and the true signal bigger than 2σ is rather unlikely. Therefore, the threshold r for filter application is chosen as $r = 2\sigma$. As the noise standard deviation σ of the TOF data depends on the signal amplitude, we need to choose r depending on the correct σ for each pixel based on the TOF amplitude image. The latter is additionally generated by the TOF camera giving information about the received relevant signal strength. This observation leads to the rule that if a pixel's distance has a margin of more than r to both borders; it is not suspect to RAN. This procedure is visualized in Fig. 6.4.

For TOF amplitude and σ comparison during PACS runtime, we generated a generic TOF camera-specific standard deviation look-up table (LUT) based on a priori measurements. Fig. 6.5 shows the relation between the measured amplitude of the TOF signal and the standard deviation (σ) of the TOF depth noise. Therein, the crosses represent σ amplitudes pairs acquired from a dedicated measurement using a plane wall target. To obtain a dense data set in the most relevant amplitude range, an arbitrary real scene is additionally evaluated, see the triangles in Fig. 6.5. To derive a generic σ-LUT, we finally use the upper envelope estimate of both data sets which is shown as solid line. The σ-LUT results obtained in this step are in line with fundamental results presented by Lange et al. [188].

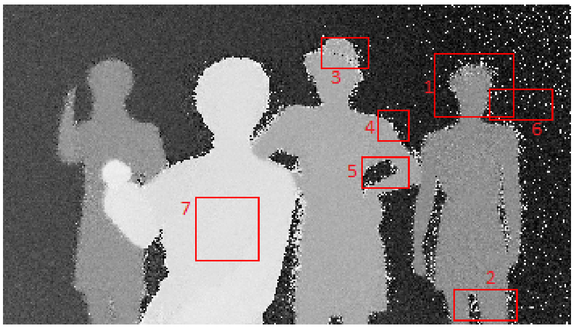


Figure 6.6: The red rectangles show the selected patches: 1 - Head, 2 - Legs, 3 - Hat, 4 - Shoulder, 5 - Arm, 6 - Wall and 7 - Doll.

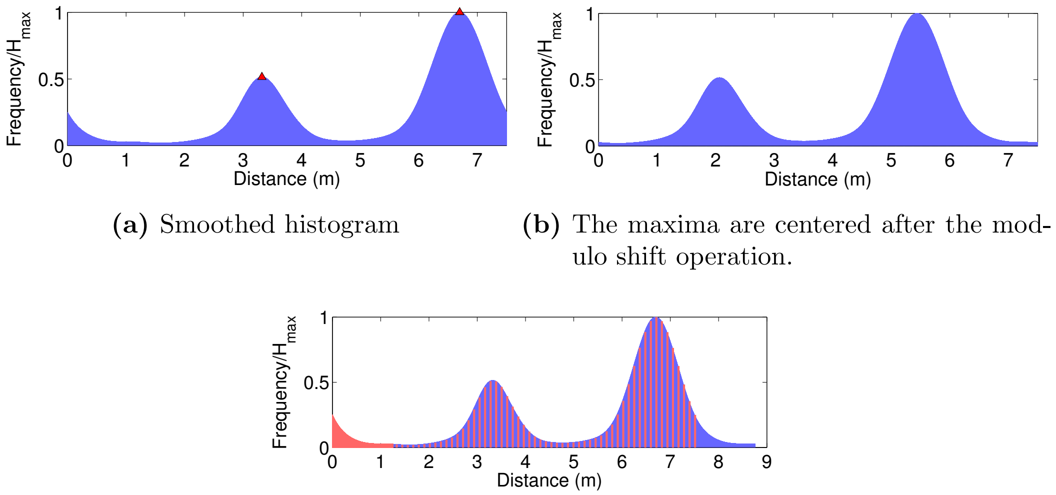
6.5 Experiments

To evaluate our proposed method a reliable ground truth is needed. We thus first explain how the ground truth is generated and subsequently present the evaluation results.

6.5.1 Generation of Ground Truth Data

A ground truth can be obtained by two approaches. First, a simulation of the data can be performed using depth image data generated artificially. Second, a real sequence can be used and the ground truth can be generated based on the real data taking a mean over a sufficiently long sequence [179]. Both approaches are not perfect: the simulated data has different characteristics compared to the real data. Especially the complex noise, shadowing and the influence of the optics are difficult to simulate. As we need realistic noise characteristics to prove the effectiveness of our approach, simulation cannot be used here. Using real camera images still has two disadvantages. First, it is limited to static scenes, because the noise is eliminated using the redundant information in other frames. Second, the described phase jumps would cause a bias error in the mean image and thus this approach is insufficient for TOF data with RAN. While we cannot overcome the first issue and thus use static sequences for our evaluation, we found a more sophisticated method to generate the reference data out of the sequence data.

Our method generates a ground truth for image patches by optimizing the non-ambiguity range such that RAN is avoided. To that end, we detect the maximum of the depth distribution and shift it such that the peak is far from the non-ambiguity range borders. The prerequisite for our method is the selection of patches. Figure 6.6 shows the seven selected patches for generating reference patches. Each patch, except the patch "Doll", contains RAN. The steps for generating reference patches are



(a) Smoothed histogram

(b) The maxima are centered after the modulo shift operation.

(c) Final histogram (blue) and original smoothed histogram (red)

Figure 6.7: The histogram of the depth values in one reference patch ("Head"). Shown is the histogram after step 1., step 3. and step 4.. The non-ambiguity range, which corresponds to the input signal range, is 7.5 m here. After the proposed transformation this range is extended and the distibutions can be assumed to be Gaussian.

1. smoothing the histogram of the patch using a Gaussian kernel,
2. searching for local maxima,
3. center the maxima using the modulo shift operation,
4. extending the domain and shifting back to the original position and
5. averaging.

The first step is necessary to improve the detection of the local maxima and therewith the depth of the objects. Figure 6.7a shows the smoothed histogram of the patch "Head". The red markers show the location of the detected maxima. The histogram shows three accumulations at around 0m, 3.3m and 6.7m. Since we know that there are just two possible depth levels in this patch (i.e., the head and the wall) relating the objects to the accumulations is simple: the accumulation at 3.3m is the head of the doll and 6.7m is the location of the wall in the background. While the accumulation at 0m pretends to be another object, it is only the distribution tail of the wall's depth values. To center the distribution, we apply a translation in combination with a modulo operation. The translation t is based on the location of the maxima X_m and connects the distribution tails close to the borders to the corresponding main part of the distribution.

Table 6.1: RMSE (m) of the static sequences by comparison. TOF represents the original data. While TJBF ($\sigma = 10$) is optimized for a static scene, TJBF ($\sigma = 0.01$) is adopted for a dynamic scene.

Sequence	TOF	Preprocessing		TJBF	
		PACS	Median	without PACS	with PACS
Head	1.354	0.738	**0.336**	1.433	**0.266**
Legs	1.387	0.836	**0.625**	0.548	**0.258**
Hat	1.121	0.748	**0.453**	1.122	**0.250**
Shoulder	1.287	0.731	**0.576**	1.371	**0.309**
Arm	1.298	0.980	**0.784**	0.932	**0.423**
Wall	1.456	0.446	**0.212**	1.800	**0.108**
Doll	0.070	0.070	**0.023**	2.081	**0.017**
mean	1.139	0.650	**0.430**	1.327	**0.233**

$$t = \begin{cases} \frac{d_{nar}}{2} - X_m & \text{when } \exists! \ X_m \\ \frac{d_{nar}}{2} - (X_{m_L} + \lfloor \frac{X_{m_R} - X_{m_L}}{2} \rfloor) & \text{otherwise} \end{cases} \tag{6.5}$$

The first case is applied when only one maximum exists (in the patch "Wall" and "Doll"). Case two calculates the translation for the remaining patches by the location of the most left maximum X_{m_L} and the most right maximum X_{m_R}. This method reaches its limits when the distance between X_{m_L} and X_{m_R} is too large (an object that is close to the camera and an object at 7.5m). Fig. 6.7b shows the centered histogram. After centering the maxima, we perform a retranslation without the modulo operation in order to shift maxima back to their original location. Fig. 6.7c shows the difference between the final histogram (blue) and the original smoothed histogram in Fig. 6.7a (red). The histogram after the final transformation is similar to the version before the modulo shift operation, but the range is extended.

To obtain the final ground truth of one patch, we apply the described transformation and then average over the transformed patches of the static sequence.

6.5.2 Results

To evaluate the effectiveness of our approach, we use our pre-processing filter followed by a typical spatio-temporal joint-bilateral filter (TJBF) and compare it to the result achieved with TJBF without PACS. As median filtering is usually applied for images degraded by salt-and-pepper noise, we additionally include the results of a spatio-temporal median filter [177].

We optimized the parameters of TJBF with and without PACS pre-filtering for minimal RMSE on the static sequence with respect to the previously generated ground truth. The corresponding RMSE values are listed in Tab. 6.1. TJBF with

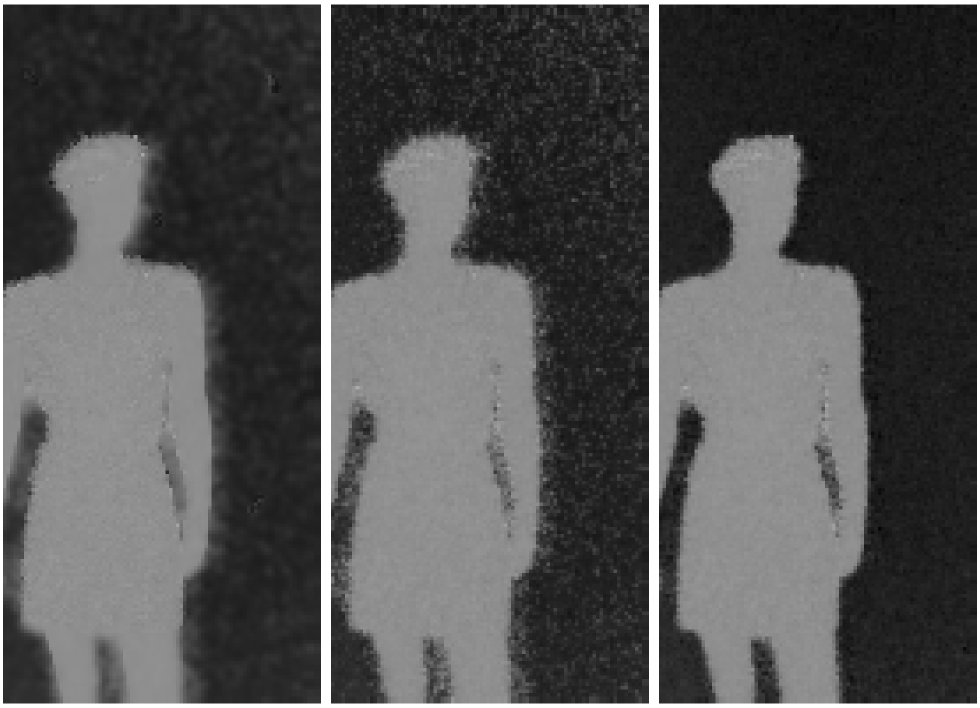


Figure 6.8: The left image is denoised by TJBF (without PACS) using the optimal parameters of a RMSE sweep with respect to the ground truth ($\sigma_S = 1, w = 4, \sigma_R = 0.01, \sigma_T = 10, temporalwindowlength = 9$). The center image (without PACS) and right image (with PACS) use TJBF optimized for usage with PACS ($\sigma_S = 0.5, w = 2, \sigma_R = 10, \sigma_T = 10, temporalwindowlength = 9$).

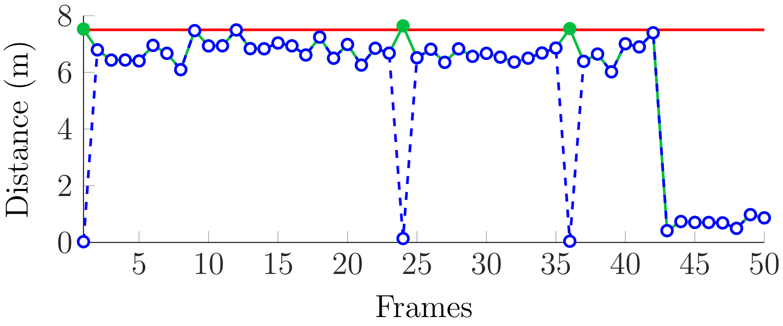


Figure 6.9: The curves illustrates the course of a pixel over 50 frames in a dynamic scene. The maximum TOF range is marked by the red line. The blue dashed line and the green line represent the TOF data and the values after PACS, respectively. An object close to the camera moves in at frame 43.

PACS achieves clearly lower RMSE values than both TJBF without PACS and the median filter.

Besides the best RMSE values, Fig. 6.8 visually proves that more details are retained: the clear outlines of the mannequin as well as the gap between left arm and body show sharp edges and high contrast. The pre-processing algorithm PACS has thus shown convincing results.

Median filtering is able to suppress RAN and achieves an average RMSE value of 0.430m on the static sequence in contrast to PACS with 0.650m. However, median filtering causes oversmoothing along edges in the depth map. This effect is prevented by PACS because PACS performs just a re-arrangement of the values instead of smoothing the depth map. This explains the allegedly better RMSE of median here. It also shows that RMSE is not able to evaluate edge preservation.

The influence of PACS on the main filter is the next evaluation step. While the denoising result of TJBF without PACS has an average RMSE of 1.327m, TJBF achieves a significant reduction to 0.258m using PACS. The observation of TJBF without PACS is again oversmoothing because of the high denoising attempt which is necessary to eliminate RAN.

The elimination of RAN using PACS works on static sequences as well as on dynamic sequences which is shown in Fig. 6.9. Therein, a single noisy pixel with RAN is observed over time. The blue line depicts the original values of this pixel whereas the green line shows the values after PACS. While the jumps of the pixel value, which are caused by RAN, are eliminated, the signal is preserved. Notice that at the end of this sequence at frame 43, an object covered by that pixel is passing into the scene and PACS does not suppress this complex dynamic behavior.

6.6 Conclusion

This chapter proposes a novel pre-processing algorithm named PACS for time-of-flight depth map denoising. It outperforms traditional filters like the median filter in scenarios where almost the entire depth range is covered by the scene, and thus, range-ambiguity noise (RAN) observable as salt-and-pepper noise becomes visible. An evaluation method together with a custom ground truth data set has been introduced. PACS with subsequent standard temporal joint-bilateral filtering (TJBF) decreases the mean RMSE value by 46% when compared to pure median filtering and by 82% when compared to pure TJBF, while preserving most edge information. Therefore, we conclude that PACS is beneficial for all TOF denoising approaches that assume underlying Gaussian noise in the raw data.

7 Conclusion

State-of-the-art denoising methods are limited in their application to real camera data, due to the unrealistic noise model assumed in the design of the algorithms and due to the misleading use of error metrics, which do not accurately reflect the human perception of visual quality. Furthermore, denoising algorithms have become more and more complex, and finding a computationally efficient solution that provides high visual quality is challenging. In the following, the main findings regarding the four research questions raised in the introduction are summarized.

How can denoising methods be improved for real camera data?

As mentioned above, the commonly assumed AWGN is fundamentally different from camera noise. In the readily processed image for display, the noise is spatially correlated, signal-dependent, and the distribution is not Gaussian. The noise of the sensor data is easier to model, but the signal properties are different from the readily processed data as the values are linear to the amount of light. High brightness differences in linear data lead to extremely steep edges, and hence, large pixel-to-pixel differences. As such large differences are rare in readily processed images, many algorithms show artifacts around edges from very bright to very dark when used on linear data. As discussed in Chapter 2, preprocessing the linear data can only reduce these artifacts for some of the algorithms.

Therefore, Chapter 3 proposes two new denoising methods for real camera data. Both methods are specifically designed to work in the linear domain: the first method operates on raw CFA data, i.e., Bayer data; and the second operates on linear RGB data, taking the spatial correlation of the camera noise into account.

The first method, presented in Sec. 3.1, relies on a shape-adaptive DCT (SA-DCT). The SA-DCT can adapt well to linear data, as the shapes are homogenous and thus do not include large pixel-to-pixel differences. Whereas the original SA-DCT was proposed for standard non-CFA data, the method proposed in this dissertation can denoise Bayer data directly. The method consists of three steps: a luminance transformation of the Bayer data, determining an adequate neighborhood for denoising, and hard thresholding in the SA-DCT domain. The method adapts to the camera noise in the raw data with a locally adaptive threshold and signal-dependent weights. Applied to real camera data and realistically simulated test data, the proposed method shows good results in a visual and in a quantitative comparison to other state-of-the-art methods. To improve the denoising results on video data, an additional temporal filtering step has been added in order to avoid temporal flickering.

Sec. 3.2 proposes a new method based on wavelet thresholding with a shrinkage function that is adapted for linear RGB data and can handle the large gradients around highlights. The method is designed for linear, debayered data and thus needs to take the spatially correlated noise into account. The method copes with the camera noise characteristics via level-dependent noise gain factors. Both the visual quality inspection and objective quality metrics show the advantages of the proposed method. As wavelet representations are widely used for denoising, the developed noise gain factors could be used to improve other wavelet-based algorithms as well.

Both proposed methods show visually appealing results. The wavelet method on linear RGB data removes more noise than the SA-DCT Bayer denoising, which operates on the four subchannels independently limiting denoising strength, as independent denoising of the color channels can lead to color artifacts. The wavelet NGF method, in contrast, operates on full-color data and higher denoising strength can be achieved. However, the remaining noise is not as natural as the remaining noise of the SA-DCT.

Which denoising methods can be implemented efficiently?

Both proposed denoising methods are designed for very low computational complexity to enable real-time processing in embedded systems. The SA-DCT can be implemented similarly to the standard block DCT, and as most compression algorithms use a DCT, efficient hardware implementations are already available. Despite memory requirements are low compared to nonlocal algorithms, the overlapping shapes require highly parralell implementation that may exceed available processing power.

The wavelet based bivariate thresholding proposed in Sec. 3.2 is based on the easy-to-compute shift-invariant DWT using low complexity filters. Despite the simple transformation, high-quality results can be achieved. While results certainly could be improved with better filters or more complex transformations, e.g., a dual-tree or stationary transformation, computational complexity would increase dramatically because the wavelet decomposition accounts for the main part of the required resources. The resource requirements are scalable with the number of wavelet decomposition levels. An implementation of three levels on an FPGA allows for real-time performance on HD data with 30 fps. In a camera, denoising is only one of many image processing tasks, and the resources needed for the wavelet decomposition might still be too extensive in such systems. However, wavelet decomposition can also be helpful for other image processing tasks, e.g., multiscale algorithms in computer vision and thus be worth the implementation effort.

For both methods, the trade-off between visual quality and memory requirements is critical, as larger neighborhood and higher number of wavelet decomposition levels enables higher visual quality. The SA-DCT requires additional buffer for storing the overlapping estimates, and hence, it can be advantageous to use the wavelet NGF method for achieving high noise reduction. However, the SA-DCT results are more natural. The trade-off is thus influenced by the processing power and memory resources available in the system, whether buffer can be shared with other image processing modules, and the visual quality goals in the application.

Combining the spatial denoising with a temporal denoising step significantly improves the results of spatial denoising methods, and hence, if an additional frame buffer is affordable in the system, temporal denoising should be included.

How can the parameters of denoising methods be adjusted to achieve the best visual quality?

To achieve the best visual quality, the denoising strength must be optimized for each image. In most publications, the parameters regulating denoising strength are optimized for high PSNR on each test image. This is not applicable in real applications, as a reference image is not available. Additionally, the PSNR does not correlate well with human perception of visual quality. Therefore, Chapter 4 proposes a method for automatic parameter optimization. The proposed approach compares the gradient histogram of a denoised image to a reference gradient histogram, which is estimated based on down- and upsampling of the noisy image. Thus, our method works without a reference and is image-adaptive. Both the results of a subjective test with 20 participants and the comparison with full-reference quality metrics show that the method estimates parameters very well: the method yields a higher image quality than existing no-reference metrics.

The method requires only very limited resources and is easy to implement. Thus, it is expected to be very attractive for real-time applications.

How can denoising algorithms be evaluated adequately and what are the important factors that influence human vision?

Realistic evaluation of denoising approaches is difficult; evaluation is mostly based on an unrealistic noise model or unrealistic test data – or even both. Therefore, Chapter 5 proposes two strategies for an improved evaluation of denoising algorithms on video data.

Sec. 5.1 shows a method to enable a more realistic evaluation of denoising results by including the full processing pipeline of a camera. To identify the properties of the camera noise that influence the perceived visual quality of noisy and denoised image sequences, a subjective test with 18 participants was performed. The test results show that signal-dependence and spatial correlation of the camera noise significantly influence the visual quality. The spatially correlated noise – introduced in the debayering step – is significantly more disturbing than AWGN. Consequently, including a realistic noise model is important for future research.

To estimate the visual quality of denoising results without time-consuming subjective tests, reliable quality metrics are required. Therefore, five no-reference metrics and 12 full-reference metrics were compared to the visual quality scores acquired by the above mentioned subjective test. The comparison shows that none of the tested metrics fully reflects the perceived visual quality. We thus need to improve quality metrics for reliable denoising evaluation.

Understanding human perception of noise is critical to improve video quality metrics for denoising evaluation. Therefore, Sec. 5.2 studies the visibility of noise based on a subjective test. For the test, noise was first filtered with a bandpass of different center frequencies, and then added to one frame of a video. For each frame, new noise with the same characteristics was generated. This noise video was then shown to the observers, first as an independent frame and second as a video. In this way, we can evaluate the perception of the same noise on a still image and on a video.

The test results show that the contrast sensitivity to spatially low frequency noise is significantly higher than for high spatial frequencies, which matches well previous results with sine waves. Whereas this tendency is visible in both video data and still images, the sensitivity difference is clearly higher in video data. Hence, low frequency noise is more visible in video data than in a still image. As expected, the results explain the above-mentioned quality drop in image sequences with spatially correlated noise compared to AWGN and the high impairment through low-frequency noise that remains after spatial denoising in videos.

Furthermore, these results indicate that metrics designed for still images should not be used directly for quality estimation of noisy and denoised videos. For this reason, sensitivity compensation factors are presented in Sec. 5.2 that can be included into future quality metrics to improve visual quality estimations for noisy and denoised videos.

Is denoising dead?

Despite the huge body of research that has already been carried out in this field, the above raised questions show that there is room for improvement. This dissertation has shown that significant improvement can be achieved accounting for a realistic camera noise model. Whereas a DCT-based and a wavelet-based method are adapted to camera noise, many other types of denoising methods exist. Similar to the noise gain factors for wavelet denoising, efficient methods for camera data may be also be found based on other denoising methods.

Evaluating denoising methods realistically and according to human perception is critical for achieving high visual quality. This dissertation proposes a low-cost method for parameter tuning, which, however, does not claim to serve as a general quality metric. Sec. 5.2 shows the significant differences in the human perception of noise in video sequences compared to still images, which indicates that, despite the vast amount of research on human vision, this research might still be advanced to tackle today's image processing problems. A quality metric reflecting human perception would allow for rapid advances in denoising development and facilitate the trade-off visual quality vs. implementation resources. Hence, denoising is not dead, and many advances can be expected in future research.

Publications by the author

[1] T. Seybold, M. Plichta, and W. Stechele. Improving wavelet denoising based on an in-depth analysis of the camera color processing. In *Proc. SPIE 9400, Real-Time Image and Video Processing 2015*, volume 9400, pages 94000Q–94000Q–13, 2015.

[2] T. Hach, T. Seybold, and H. Böttcher. Phase-aware candidate selection for time-of-flight depth map denoising. volume 9393, pages 93930E–93930E–9, 2015.

[3] T. Seybold, B. Klässner, and W. Stechele. Chapter 1 - Denoising camera data: Shape-adaptive noise reduction for color filter array image data. In L. Deligiannidis and H. Arabnia, editors, *Emerging Trends in Image Processing, Computer Vision and Pattern Recognition*, pages 3–17. Morgan Kaufmann, Boston, 2015.

[4] T. Seybold, F. Kuhn, J. Habigt, M. Hartenstein, and W. Stechele. Automatic denoising parameter estimation using gradient histograms. In *Visual Communications and Image Processing Conference, 2014 IEEE*, pages 358–361. IEEE, 2014.

[5] T. Seybold, Ö. Cakmak, C. Keimel, and W. Stechele. Noise characteristics of a single sensor camera in digital color image processing. *Color and Imaging Conference*, 2014(22):53–58, November 2014.

[6] Adapting the bivariate denoising method for real camera noise. In *20. Workshop Farbbildverarbeitung, 2014 German Color Group.*

[7] T. Seybold, B. Klässner, and W. Stechele. Denoising camera data: Shape-adaptive noise reduction for color filter array image data. In *The 2014 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2014)*, July 2014.

[8] T. Seybold, M. Knopp, C. Keimel, and W. Stechele. Beyond standard noise models: Evaluating denoising algorithms with respect to realistic camera noise. *International Journal of Semantic Computing*, 08(02):145–167, June 2014.

[9] M. Moeller, E. Brinkmann, M. Burger, and T. Seybold. Color Bregman TV. *SIAM Journal on Imaging Sciences*, 7(4):2771–2806, January 2014.

[10] T. Seybold, C. Keimel, M. Knopp, and W. Stechele. Towards an evaluation of denoising algorithms with respect to realistic camera noise. In *Proceedings of the 2013 IEEE International Symposium on Multimedia*, pages 203–210, 2013.

[11] S. Andriani, H. Brendel, T. Seybold, and J. Goldstone. Beyond the kodak image set: A new reference set of color image sequences. In *ICIP*, pages 2289–2293, 2013.

Bibliography

[12] J. Aelterman, B. Goossens, A. Pi\vzurica, and W. Philips. Locally adaptive complex wavelet-based demosaicing for color filter array images. In *IS&T/SPIE Electronic Imaging*, pages 72480J–72480J, 2009.

[13] J. Aelterman, B. Goossens, A. Pizurica, and W. Philips. Suppression of Correlated Noise. In A. Zaher, editor, *Recent Advances in Signal Processing*. InTech, November 2009.

[14] Michael Aharon, Michal and and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, November 2006.

[15] Stefano Andriani and Harald Brendel. Crosstalk correction technique for single sensor camera provided with Bayer color filter array. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 2252–2255. IEEE, 2013.

[16] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*. Springer, 2006.

[17] M. Benning, C. Brune, M. Burger, and J. Müller. Higher-order TV methods—enhancement via Bregman iteration. *J. Sci. Comput.*, 54(2-3):269–310, 2013.

[18] P. Blomgren and T. F. Chan. Color TV: Total variation methods for restoration of vector valued images. *IEEE Trans. Image Processing*, 7:304–309, 1996.

[19] X. Bresson and T. F. Chan. Fast dual minimization of the vectorial total variation norm and applications to color image processing. *Inverse Problems and Imaging*, 2(4):255–284, 2008.

[20] A. Buades, B. Coll, and J. Morel. A review of image denoising algorithms, with a new one. *Multiscale Modeling & Simulation*, 4(2):490 – 530, 2005.

[21] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. Nonlocal Image and Movie Denoising. *International Journal of Computer Vision*, 76(2):123–139, February 2008.

[22] Peter D. Burns and Roy S. Berns. Error propagation analysis in color measurement and imaging. *Color Res. Appl.*, 22(4):280–289, August 1997.

[23] T. Chan and J. Shen. *Image Processing and Analysis.* SIAM, Philadelphia, 2005.

[24] D. M Chandler and S. S Hemami. VSNR: a wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16(9):2284 – 2298, 2007.

[25] P. Chatterjee, N. Joshi, S. B. Kang, and Y. Matsushita. Noise suppression in low-light images through joint denoising and demosaicing. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 321–328. IEEE, 2011.

[26] P. Chatterjee and P. Milanfar. Practical Bounds on Image Denoising: From Estimation to Information. *IEEE Transactions on Image Processing*, 20(5):1221–1233, May 2011.

[27] Priyam Chatterjee and Peyman Milanfar. Fundamental limits of image denoising: are we there yet? In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1358–1361. IEEE, 2010.

[28] P. Chatterjee and P. Milanfar. Is Denoising Dead? *IEEE Transactions on Image Processing*, 19(4):895–911, April 2010.

[29] P. Chatterjee and P. Milanfar. Clustering-Based Denoising With Locally Learned Dictionaries. *IEEE Transactions on Image Processing*, 18(7):1438–1451, 2009.

[30] Guangquan Cheng, Jincai Huang, Zhong Liu, and Cheng Lizhi. Image quality assessment using natural image statistics in gradient domain. *AEU - International Journal of Electronics and Communications*, 65(5):392–397, May 2011.

[31] Taeg Sang Cho, Neel Joshi, C. Lawrence Zitnick, Sing Bing Kang, Richard Szeliski, and William T. Freeman. A content-aware image prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 169 – 176. IEEE, 2010.

[32] Taeg Sang Cho, C. L. Zitnick, N. Joshi, Sing Bing Kang, R. Szeliski, and W. T. Freeman. Image restoration by matching gradient distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):683–694, April 2012.

[33] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080 – 2095, 2007.

[34] G. Dillen, B. Georis, J. Legat, and O. Cantineau. Combined line-based architecture for the 5-3 and 9-7 wavelet transform of jpeg2000. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(9):944–950, 2003.

[35] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli. New full-reference quality metrics based on HVS. In *VPQM*, 2006.

[36] M. Elad and M. Aharon. Image denoising via learned dictionaries and sparse representation. In *CVPR*, volume 1, pages 895 – 900, 2006.

[37] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image Processing*, 15(12):3736–3745, 2006.

[38] A. Foi. Clipped noisy images: heteroskedastic modeling and practical denoising. *Signal Processing*, pages 2609 – 2629, 2009.

[39] Alessandro Foi. Practical denoising of clipped or overexposed noisy images. In *Proc. 16th Eur. Signal Process. Conf., EUSIPCO 2008*, 2008.

[40] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. *IEEE Transactions on Image Processing*, 16(5):1395 – 1411, may 2007.

[41] A. Foi, V. Katkovnik, and K. Egiazarian. Signal-dependent noise removal in pointwise shape-adaptive DCT domain with locally adaptive variance. In *EUSIPCO*, 2007.

[42] Bernd Girod. What's Wrong With Mean Squared Error? In Andrew B. Watson, editor, *Digital Images and Human Vision*, pages 207–220. MIT Press, Cambridge, MA, USA, 1993.

[43] R. Giryes and M. Elad. Sparsity based poisson denoising. In *2012 IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI)*, pages 1 – 5, nov. 2012.

[44] B. Goldluecke and D. Cremers. An approach to vectorial total variation based on geometric measure theory. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 327–333, 2010.

[45] Gene H. Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215, May 1979.

[46] B. Goossens, J. Aelterman, H. Luong, A. Pizurica, and W. Philips. Complex wavelet joint denoising and demosaicing using Gaussian scale mixtures. In *2013 20th IEEE International Conference on Image Processing (ICIP)*, pages 445–448, September 2013.

[47] B.K. Gunturk, Y. Altunbasak, and R.M. Mersereau. Color plane interpolation using alternating projections. *IEEE Transactions on Image Processing*, 11(9):997–1013, September 2002.

[48] B.K. Gunturk, J. Glotzbach, Y. Altunbasak, R.W. Schafer, and R.M. Mersereau. Demosaicking: color filter array interpolation. *Signal Processing Magazine, IEEE*, 22(1):44–54, 2005.

[49] H.R. Sheikh, Z.Wang, L. Cormack and A.C. Bovik. LIVE image quality assessment database release 2.

[50] Per Christian Hansen and Dianne Prost O'Leary. The use of the l-curve in the regularization of discrete ill-posed problems. *SIAM Journal on Scientific Computing*, 14(6):1487–1503, November 1993.

[51] K. Hirakawa, X.-L. Meng, and P.J. Wolfe. A framework for wavelet-based analysis and processing of color filter array images with applications to denoising and demosaicing. In *ICASSP*, volume 1, pages I–597 – I–600, 2007.

[52] K. Hirakawa and T.W. Parks. Joint demosaicing and denoising. *IEEE Transactions on Image Processing*, 15(8):2146–2157, August 2006.

[53] Chih-Hsien Hsia, Jen-Shiun Chiang, and Jing-Ming Guo. Memory-efficient hardware architecture of 2-d dual-mode lifting-based discrete wavelet transform. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(4):671–683, April 2013.

[54] Jinggang Huang and D. Mumford. Statistics of natural images and models. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 541–547 Vol. 1, 1999.

[55] IEC. IEC/4WD 61966-2-1: Colour measurement and management in multimedia systems and equipment - part 2.1: Default colour space - sRGB., May 2006.

[56] ITU. Rec. ITU-R BT.709-4: Parameter values for the HDTV standards for production and international programme exchange, Mar 2000.

[57] G. Jeon and E. Dubois. Demosaicking of noisy bayer-sampled color images with least-squares luma-chroma demultiplexing and noise level estimation. *IEEE Transactions on Image Processing*, 22(1):146–156, 2013.

[58] G. Jeon and E. Dubois. Demosaicking of Noisy Bayer-Sampled Color Images With Least-Squares Luma-Chroma Demultiplexing and Noise Level Estimation. *IEEE Transactions on Image Processing*, 22(1):146–156, January 2013.

[59] Don H. Johnson and Sinan Sinanovic. Symmetrizing the kullback-leibler distance. Technical report, Rice University, 2001.

[60] Mihajlo Katona, Aleksandra Pižurica, Nikola Teslić, Vladimir Kovačević, and Wilfried Philips. Fpga design and implementation of a wavelet-domain video denoising system. In Jacques Blanc-Talon, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, volume 3708 of *Lecture Notes in Computer Science*, pages 650–657. Springer Berlin Heidelberg, 2005.

[61] Mihajlo Katona, Aleksandra Pižurica, Nikola Teslić, Vladimir Kovačević, and Wilfried Philips. A real-time wavelet-domain video denoising implementation in fpga. *EURASIP J. Embedded Syst.*, 2006(1):6–6, January 2006.

[62] M. Katona, A. Pižurica, N. Teslić, V. Kovačević, and W. Philips. A real-time wavelet-domain video denoising implementation in FPGA. *EURASIP J. Embedded Syst.*, 2006(1), January 2006.

[63] D. Khashabi, S. Nowozin, J. Jancsary, and A. W. Fitzgibbon. Joint Demosaicing and Denoising via Learned Nonparametric Random Fields. *IEEE Transactions on Image Processing*, 23(12):4968–4981, December 2014.

[64] A. Kokaram, D. Kelly, H. Denman, and A. Crawford. Measuring noise correlation for improved video denoising. In *ICIP*, pages 1201–1204, 2012.

[65] E. C. Larson and D. M. Chandler. Most apparent distortion: full-reference image quality assessment and the role of strategy. *Journal of Electronic Imaging*, 19(1):011006, 2010.

[66] M. Lebrun, A. Buades, and J. Morel. A Nonlocal Bayesian Image Denoising Algorithm. *SIAM J. Imaging Sci.*, 6(3):1665–1688, January 2013.

[67] M. Lebrun, M. Colom, Antoni Buades, and J. M. Morel. Secrets of image denoising cuisine. *Acta Numerica*, 21:475–576, 2012.

[68] Jong-Sen Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics, and Image Processing*, 24(2):255–269, November 1983.

[69] A. Levin and B. Nadler. Natural image denoising: Optimality and inherent bounds. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2833–2840, 2011.

[70] Chaofeng Li, A C Bovik, and Xiaojun Wu. Blind image quality assessment using a general regression neural network. *IEEE Transactions on Neural Networks*, 22(5):793–799, May 2011.

[71] X. Li, B. Gunturk, and L. Zhang. Image demosaicing: a systematic survey. volume 6822, pages 68221J–68221J–15, 2008.

[72] Chung-Jr Lian, Kuan-Fu Chen, Hong-Hui Chen, and Liang-Gee Chen. Lifting based discrete wavelet transform architecture for jpeg2000. In *Circuits and Systems, 2001. ISCAS 2001. The 2001 IEEE International Symposium on*, volume 2, pages 445–448 vol. 2, 2001.

[73] M Lindenbaum, M Fischer, and A Bruckstein. On Gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, January 1994.

[74] Ingmar Lissner, Jens Preiss, Philipp Urban, Matthias Scheller Lichtenauer, and Peter Zolliker. Image-difference prediction: From grayscale to color. *IEEE Transactions on Image Processing*, 22(2):435–446, February 2013.

[75] C. Liu and W. Freeman. A high-quality video denoising algorithm based on reliable motion estimation. *Computer Vision – ECCV 2010*, pages 706–719, 2010.

[76] C. Liu, R. Szeliski, S. B. Kang, C.L. Zitnick, and W.T. Freeman. Automatic estimation and removal of noise from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):299 – 314, 2008.

[77] W. Lu and Y.-P. Tan. Color filter array demosaicking: new method and performance measures. *IEEE Transactions on Image Processing*, 12(10):1194 – 1210, oct. 2003.

[78] V. Lukin, S. Abramov, N. Ponomarenko, K. Egiazarian, and J. Astola. Image filtering: Potential efficiency and current problems. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1433–1436, May 2011.

[79] M.Burger and S.Osher. A guide to the tv zoo. In M.Burger and S.Osher, editors, *Level Set and PDE -based Reconstruction Methods in Imaging*, pages 1–70. Springer, Berlin, 2013.

[80] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian. Video Denoising, Deblocking, and Enhancement Through Separable 4-D Nonlocal Spatiotemporal Transforms. *IEEE Transactions on Image Processing*, 21(9):3952–3966, 2012.

[81] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *ICCV*, pages 2272 – 2279, 2009.

[82] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Signal Processing*, 17(1):53–69, January 2008.

[83] D. Menon and G. Calvagno. Joint demosaicking and denoisingwith space-varying filters. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 477–480, 2009.

[84] D. Menon and G. Calvagno. Regularization Approaches to Demosaicking. *IEEE Transactions on Image Processing*, 18(10):2209–2220, October 2009.

[85] D. Menon and G. Calvagno. Demosaicing Based Onwavelet Analysis of the Luminance Component. In *IEEE International Conference on Image Processing, 2007. ICIP 2007*, volume 2, pages II – 181–II – 184, 2007.

[86] S. Milani and G. Calvagno. Joint denoising and interpolation of depth maps for MS Kinect sensors. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 797–800, 2012.

[87] A. Mittal, A. K. Moorthy, and A. C. Bovik. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21(12):4695–4708, December 2012.

[88] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, March 2013.

[89] Y. Monno, D. Kiku, S. Kikuchi, M. Tanaka, and M. Okutomi. Multispectral Demosaicking with Novel Guide Image Generation and Residual Interpolation. In *ICIP2014*, pages 645–649, 2014.

[90] A. K Moorthy and A. C Bovik. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, 20(12):3350–3364, December 2011.

[91] Anush Krishna Moorthy and Alan Conrad Bovik. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17(5):513–516, 2010.

[92] D. D. Muresan and T. W. Parks. Adaptive principal components and image denoising. In *ICIP (1)*, pages 101–104, 2003.

[93] Usha Bhanu Nageswaran and A. Chilambuchelvan. High speed vlsi implementation of lifting based dwt. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, ICACCI ’12, pages 1104–1110. ACM, 2012.

[94] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *SIAM Multiscale Model. Simul.*, 4:460–489, 2005.

[95] D. Paliy, A. Foi, R. Bilcu, and V. Katkovnik. Denoising and interpolation of noisy bayer data with adaptive cross-color filters. *Proc. of SPIE-IS&T EI, VCIP*, 2008.

[96] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian. Spatially Adaptive Color Filter Array Interpolation for Noiseless and Noisy Data. *Int. J. Imaging Syst. Technol.*, 17(3):105–122, October 2007.

[97] S. H. Park, H. S. Kim, S. Lansel, M. Parmar, and B.A. Wandell. A case for denoising before demosaicking color filter array data. In *Conference Record of the 43rd Asilomar Conference on Signals, Systems and Computers*, pages 860 – 864, 2009.

[98] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, July 1990.

[99] Nikolay Ponomarenko, Oleg Ieremeiev, Vladimir Lukin, Karen Egiazarian, and Marco Carli. Modified image visual quality metrics for contrast change and mean shift accounting. In *Proceedings of CADSM*, pages 305–311, 2011.

[100] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti. TID2008-A database for evaluation of full-reference visual quality assessment metrics. *Advances of Modern Radioelectronics*, 10(10):30 – 45, 2009.

[101] J. Portilla, V. Strela, M. J Wainwright, and E. P Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *IEEE Transactions on Image Processing*, 12(11):1338 – 1351, 2003.

[102] S. Ramani, T. Blu, and M. Unser. Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Transactions on Image Processing*, 17(9):1540–1554, September 2008.

[103] Jörg Ritter and Paul Molitor. A pipelined architecture for partitioned dwt based lossy image compression using fpga's. In *Proceedings of the 2001 ACM/SIGDA Ninth International Symposium on Field Programmable Gate Arrays*, FPGA '01, pages 201–206. ACM, 2001.

[104] L. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

[105] M. A. Saad, A. C. Bovik, and C. Charrier. Blind image quality assessment: A natural scene statistics approach in the DCT domain. *IEEE Transactions on Image Processing*, 21(8):3339–3352, August 2012.

[106] Michele A Saad, Alan C Bovik, and Christophe Charrier. A DCT statistics-based blind image quality index. *IEEE Signal Processing Letters*, 17(6):583–586, June 2010.

[107] J. Salvador, M. Borsum, and A. Kochale. A Bayesian approach for natural image denoising. In *2013 20th IEEE International Conference on Image Processing (ICIP)*, pages 1095–1099, September 2013.

[108] G. Sapiro and D.L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Trans. Img. Proc.*, 5(11):1582–1586, 1996.

[109] M. Schöberl, W. Schnurrer, A. Oberdörster, S. Fößel, and A. Kaup. Dimensioning of optical birefringent anti-alias filters for digital cameras. In *ICIP*, pages 4305 – 4308, 2010.

[110] I. W. Selesnick, R. G. Baraniuk, and N. G. Kingsbury. The dual-tree complex wavelet transform - a coherent framework for multiscale signal and image processing. *IEEE Signal Processing Magazine*, 22(6):123–151, 2005.

[111] L. Sendur and I.W. Selesnick. Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. *IEEE Transactions on Signal Processing*, 50(11):2744–2756, November 2002.

[112] H.R. Sheikh and A.C. Bovik. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, Feb 2006.

[113] H.R. Sheikh, A.C. Bovik, and G. de Veciana. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing*, 14(12):2117 – 2128, 2005.

[114] H.R. Sheikh, M.F. Sabir, and A.C. Bovik. A statistical evaluation of recent full reference image quality assessment algorithms. *IEEE Transactions on Image Processing*, 15(11):3440 – 3451, 2006.

[115] L. Stroebel, J. Compton, I. Current, and R. Zakia. Tone reproduction. In *Basic Photografic Materials and Processes*, pages 235–255. Elsevier, 2000.

[116] J. Suo, Y. Deng, L. Bian, and Q. Dai. Joint Non-Gaussian Denoising and Superresolving of Raw High Frame Rate Videos. *IEEE Transactions on Image Processing*, 23(3):1154–1168, March 2014.

[117] H. Talebi and P. Milanfar. Global denoising is asymptotically optimal. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 818–822, October 2014.

[118] M. Teutsch, P. Trantelle, and J. Beyerer. Adaptive real-time image smoothing using local binary patterns and Gaussian filters. In *2013 20th IEEE International Conference on Image Processing (ICIP)*, pages 1120–1124, September 2013.

[119] Maria A. Trenas, Juan Lbpez, and Emilio L. Zapata. Fpga implementation of wavelet packet transform with reconfigurable tree structure. In *Euromicro Conference, 2000. Proceedings of the 26th*, pages 244–251 vol.1, 2000.

[120] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53:4655–4666, 2007.

[121] H. J. Trussell and R. Zhang. The dominance of poisson noise in color digital cameras. In *ICIP*, pages 329 – 332, 2012.

[122] E. Vansteenkiste, D. Weken, W. Philips, and E. Kerre. Perceived image quality measurement of state-of-the-art noise reduction schemes. In *Advanced Concepts for Intelligent Vision Systems*, pages 114 – 126. Springer Berlin Heidelberg, 2006.

[123] Z. Wang and A.C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98 – 117, 2009.

[124] Z. Wang, A. C Bovik, H. R. Sheikh, and E. P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600 – 612, April 2004.

[125] Zhou Wang and Qiang Li. Information content weighting for perceptual image quality assessment. *IEEE Transactions on Image Processing*, 20(5):1185–1198, May 2011.

[126] Z. Wang, E. P Simoncelli, and A. C Bovik. Multiscale structural similarity for image quality assessment. In *Signals, Systems and Computers, 2003. Conference Record of the 37th Asilomar Conference on*, volume 2, pages 1398 – 1402, 2003.

[127] J. Yang, J. Wright, T. Huang, and Y Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

[128] L. Zhang, W. Dong, D. Zhang, and G. Shi. Two-stage image denoising by principal component analysis with local pixel grouping. *Pattern Recognition*, 43(4):1531–1549, April 2010.

[129] L. Zhang, R. Lukac, X. Wu, and D. Zhang. PCA-Based spatially adaptive denoising of CFA images for single-sensor digital cameras. *IEEE Transactions on Image Processing*, 18(4):797 – 812, 2009.

[130] L. Zhang and X. Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing*, 14(12):2167 – 2178, 2005.

[131] L. Zhang, X. Wu, and D. Zhang. Color reproduction from noisy cfa data of single sensor digital cameras. *IEEE Transactions on Image Processing*, 16(9):2184 – 2197, 2007.

[132] X. Zhu and P. Milanfar. Automatic parameter selection for denoising algorithms using a no-reference measure of image content. *IEEE Transactions on Image Processing*, 19(12):3116 – 3132, 2010.

[133] Wangmeng Zuo, Lei Zhang, Chunwei Song, and D. Zhang. Texture enhanced image denoising via gradient histogram preservation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1203–1210, June 2013.

[134] ALEXA Color Processing White Paper. Technical report, ARRI.

[135] ALEXA Log C Curve - Usage in VFX Reference Manual. Technical report, ARRI.

[136] http://r0k.us/graphics/kodak/.

[137] ITU-R BT.500-13 methodology for the subjective assessment of the quality for television pictures, 2012.

[138] EMVA 1288, standard for characterization of image sensors and cameras, 2010.

[139] J. A. J. Roufs. Dynamic properties of vision–I. Experimental relationships between flicker and flash thresholds. *Vision Research*, 12(2):261–278, February 1972.

[140] J. A. J. Roufs. Dynamic properties of vision–II. Theoretical relationships between flicker and flash thresholds. *Vision Research*, 12(2):279–292, February 1972.

[141] J. A. J. Roufs. Dynamic properties of vision–III. Twin flashes, single flashes and flickerfusion. *Vision Research*, 13(2):309–323, February 1973.

[142] J. A. J. Roufs. Dynamic properties of vision–IV: Thresholds of decremental flashes, incremental flashes and doublets in relation to flicker fusion. *Vision Research*, 14(9):831–851, September 1974.

[143] J. A. J. Roufs. Dynamic properties of vision–V: Perception lag and reaction time in relation to flicker and flash thresholds. *Vision Research*, 14(9):853–869, September 1974.

[144] F. Ebner and M. D. Fairchild. Development and Testing of a Color Space (IPT) with Improved Hue Uniformity. *Color and Imaging Conference*, 1998(1):8–13, January 1998.

[145] S. Winkler and S. Süsstrunk. Visibility of noise in natural images. In *Electronic Imaging 2004*, pages 121–129. International Society for Optics and Photonics, 2004.

[146] S. Winkler. A Perceptual Distortion Metric for Digital Color Video. In *Electronic Imaging '99*, pages 175–184, San Jose, California, 1999.

[147] D. H. Kelly. Visual Responses to Time-Dependent Stimuli. I. Amplitude Sensitivity Measurements. *J. Opt. Soc. Am.*, 51(4):422–429, April 1961.

[148] D. H. Kelly. Theory of Flicker and Transient Responses, II. Counterphase Gratings. *J. Opt. Soc. Am.*, 61(5):632–640, May 1971.

[149] D. H. Kelly. Frequency Doubling in Visual Responses. *J. Opt. Soc. Am.*, 56(11):1628–1632, November 1966.

[150] D. H. Kelly. Motion and vision. II. Stabilized spatio-temporal threshold surface. *J. Opt. Soc. Am.*, 69(10):1340–1349, October 1979.

[151] D. H. Kelly. Spatio-temporal frequency characteristics of color-vision mechanisms. *J. Opt. Soc. Am.*, 64(7):983–990, July 1974.

[152] D. H. Kelly. Visual contrast sensitivity. *Journal of modern optics*, 24(2):107–129, 1977.

[153] D. H. Kelly. Spatiotemporal variation of chromatic and achromatic contrast thresholds. *J. Opt. Soc. Am.*, 73(6):742–749, June 1983.

[154] H. De Lange Dzn. Experiments on flicker and some calculations on an electrical analogue of the foveal systems. *Physica*, 18(11):935–950, November 1952.

[155] H. De Lange Dzn. Relationship between Critical Flicker-Frequency and a Set of Low-Frequency Characteristics of the Eye. *J. Opt. Soc. Am.*, 44(5):380–388, May 1954.

[156] H. De Lange Dzn. Research into the Dynamic Nature of the Human Fovea -¿ Cortex Systems with Intermittent and Modulated Light. I. Attenuation Characteristics with White and Colored Light. *J. Opt. Soc. Am.*, 48(11):777–783, November 1958.

[157] H. De Lange Dzn. Research into the Dynamic Nature of the Human Fovea -¿ Cortex Systems with Intermittent and Modulated Light. II. Phase Shift in Brightness and Delay in Color Perception. *J. Opt. Soc. Am.*, 48(11):784–787, November 1958.

[158] H. De Lange Dzn. Eye's Response at Flicker Fusion to Square-Wave Modulation of a Test Field Surrounded by a Large Steady Field of Equal Mean Luminance. *J. Opt. Soc. Am.*, 51(4):415–421, April 1961.

[159] J. G. Robson. Spatial and temporal contrast sensitivity functions of the visual system. *Journal of the Optical Society of America*, 56(8):1141–1142, 1966.

[160] F. L. Van Nes, J. J. Koenderine, H. Nas, and M. A. Bouman. Spatiotemporal Modulation Transfer in the Human Eye. *J. Opt. Soc. Am.*, 57(9):1082–1087, September 1967.

[161] K. T. Blackwell. The effect of white and filtered noise on contrast detection thresholds. *Vision Research*, 38(2):267–280, January 1998.

[162] R. E. Fredericksen and R. F. Hess. Temporal detection in human vision: dependence on stimulus energy. *J. Opt. Soc. Am. A*, 14(10):2557–2569, October 1997.

[163] R. E. Fredericksen and R. F. Hess. Estimating multiple temporal mechanisms in human vision. *Vision Research*, 38(7):1023–1040, April 1998.

[164] F. W. Campbell, R. H. S. Carpenter, and J. Z. Levinson. Visibility of aperiodic patterns compared with that of sinusoidal gratings. *The Journal of Physiology*, 204(2):283–298, 1969.

[165] F. W. Campbell and J. G. Robson. Application of Fourier analysis to the visibility of gratings. *The Journal of physiology*, 197(3):551–566, 1968.

[166] J. Jiang, L. Zhang, and J. Yang. Mixed Noise Removal by Weighted Encoding With Sparse Nonlocal Regularization. *IEEE Transactions on Image Processing*, 23(6):2651–2662, June 2014.

[167] Tong Sun and Yrjö Neuvo. Detail-preserving median based filters in image processing. *Pattern Recogn. Lett.*, 15(4):341–347, April 1994.

[168] A. Nieminen, P. Heinonen, and Y. Neuvo. A new class of detail-preserving filters for image processing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (1):74–90, 1987.

[169] Y. Dong and S. Xu. A New Directional Weighted Median Filter for Removal of Random-Valued Impulse Noise. *IEEE Signal Processing Letters*, 14(3):193–196, March 2007.

[170] S.-J. Ko and Y. H. Lee. Center weighted median filters and their applications to image enhancement. *Circuits and Systems, IEEE Transactions on*, 38(9):984–993, 1991.

[171] G. Pok, J.-C. Liu, and A.S. Nair. Selective removal of impulse noise based on homogeneity level information. *IEEE Transactions on Image Processing*, 12(1):85–92, January 2003.

[172] D. R. K. Brownrigg. The weighted median filter. *Communications of the ACM*, 27(8):807–818, 1984.

[173] Tao Chen, Kai-Kuang Ma, and Li-Hui Chen. Tri-state median filter for image denoising. *IEEE Transactions on Image Processing*, 8(12):1834–1838, December 1999.

[174] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[175] M. Camplani and L. Salgado. Adaptive spatio-temporal filter for low-cost camera depth maps. *International Conference on Emerging Signal Processing Applications*, pages 33–36, 2012.

[176] S.-Y. Kim, J.-H. Cho, A. Koschan, and M.A. Abidi. Spatial and Temporal Enhancement of Depth Images Captured by a Time-of-Flight Depth Sensor. *20th International Conference on Pattern Recognition*, pages 2358–2361, 2010.

[177] F.-Y. Lin, Y.-L. Wu, and W.-C. Hung. Denoising time-of-flight depth maps using temporal median filter. In *The 2014 International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV 2013)*, 2013.

[178] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM SIGGRAPH 2007 Papers*, page Article No. 96, 2007.

[179] L. Jovanov, A. Pižurica, and W. Philips. Denoising algorithm for the 3D depth map sequences based on multihypothesis motion estimation. *EURASIP Journal on Advances in Signal Processing*, 2011(1):1–17, 2011.

[180] T. Edeler. *Bildverbesserung von Time-Of-Fligh Tiefenkarten*, pages 12–15. Shaker, Aachen, 2012.

[181] S. Dolinar, D. Divsalar, J. Hamkins, and F. Pollara. Capacity of pulse-position modulation (PPM) on Gaussian and Webb channels. *JPL TMO Progress Report*, 42(142):1–31, 2000.

[182] R.H. Chan, Chung-Wa, and M. Nikolova. Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization. *IEEE Transactions on Image Processing*, 14(10):1479–1485, October 2005.

[183] B. Jutzi. Extending the range measurement capabilities of modulated range imaging devices by time-frequency-multiplexing. *AVN - Allgemeine Vermessungs-Nachrichten*, 2012(2):54–62, 2012.

[184] D. Droeschel, D. Holz, and S. Behnke. Multi-frequency phase unwrapping for time-of-flight cameras. *41st International Symposium on and 6th German Conference on Robotics*, pages 1–7, 2010.

[185] M. Hansard, S. Lee, O. Choi, and R. Horaud. *Time-of-Flight Cameras*, pages 30–33. Springer London, London, 2013.

[186] S. H. McClure, M. J. Cree, A. A. Dorrington, and A. D. Payne. Resolving depth-measurement ambiguity with commercially available range imaging cameras. *Proc. SPIE 7538 Image Processing: Machine Vision Applications III*, pages 75380K–75380K–12, 2010.

[187] H. Hwang and R. Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*, 4(4):499–502, 1995.

[188] R. Lange and P. Seitz. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics*, 37(3):390–397, 2001.