IEEE Transactions on Quantum Engineering

Received February 1, 2022; revised March 31, 2022; accepted April 4, 2022; date of publication April 8, 2022;
date of current version June 6, 2022.

Digital Object Identifier 10.1109/TQE.2022.3165968

Noise Reduction Methods for Charge Stability Diagrams of Double Quantum Dots

SARAH FLEITMANN[1], FABIAN HADER[1], JAN VOGELBRUCH[1], SIMON HUMPOHL[2,3], TOBIAS HANGLEITER[2,3], STEFANIE MEYER[1], AND STEFAN VAN WAASEN[1,4]

[1]Central Institute of Engineering, Electronics and Analytics ZEA-2—Electronic Systems, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany
[2]JARA-FIT Institute for Quantum Information, Forschungszentrum Jülich GmbH, 52425 Jülich, Germany
[3]RWTH Aachen University, 52074 Aachen, Germany
[4]Faculty of Engineering—Communication Systems, University of Duisburg-Essen, 47057 Duisburg, Germany

Corresponding author: Sarah Fleitmann (e-mail: s.fleitmann@fz-juelich.de).

This work was supported by the Impulse and Networking Fund “Scalable Solid-State Quantum Computing” of the Helmholtz Association.

ABSTRACT Operating semiconductor quantum dots as quantum bits requires isolating single electrons by adjusting gate voltages. The transitions of electrons to and from the dots appear as a honeycomb-like pattern in recorded charge stability diagrams (CSDs). Thus, detecting the pattern is essential to tune a double dot, but manual tuning is seriously time-consuming. However, automation of this process is difficult because the transitions' contrast is often low, and noise and background disorder potential shifts disturb the CSDs. Therefore, the signal-to-noise ratio needs to be increased to improve the detection of the line pattern. For this purpose, we evaluate a representative set of edge-preserving smoothing filters and compare them both quantitatively and qualitatively by suitable metrics and visual assessment. We generate artificial data to use full-reference metrics for the evaluation procedure and to optimize the filter parameters. Based on the results of this article, the methods attain a moderate to good amount of noise reduction and structure improvement dependent on the different CSD qualities. In conclusion, we suggest introducing the block-matching and three-dimensional transform-domain filter into the automated tuning processing pipeline. If the data are corrupted by significant amounts of random telegraph noise, the bilateral filter and the rolling guidance filter are also good choices.

INDEX TERMS Automated tuning, double quantum dots, edge-preserving smoothing, noise reduction, quantum computing.

I. INTRODUCTION

Quantum computing uses quantum mechanical effects for particular computing problems, promising an exponential speed-up compared with any classical computer. Among others, one can encode quantum bits (qubits) using quantum mechanical states of atoms, ions, electrons, or even photons. Due to their potential for scalability, spin qubits are a promising candidate [1]–[3].

Here, we consider gate-defined double quantum dots in a GaAs/AlGaAs heterostructure in the single electron regime. It isolates two electrons in a thin layer of electron gas, called 2-D electron gas (2DEG), that is existent in the semiconductor heterostructure. One can use this system for several qubit encodings, such as single-spin and, like in our case, singlet-triplet or as a part of a three-dot exchange only qubit. The quantum information is stored in the combined spin states of the two electrons of a double quantum dot [4], [5, Ch. 8]. In our work, we consider two different gate layouts, gate layout 1 (GL1) (adapted from [6]) in Fig. 1(a) and gate layout 2 (GL2) [7] in Fig. 1(b). Voltages applied to the gate electrodes deplete the area underneath the gates. Furthermore, they affect the dots by changing the tunnel barriers between adjacent quantum dots (barrier gates) or by shifting the electrochemical potential of a specific quantum dot (plunger gates) [8]. The experimental setup [7], [9] is identical for both layouts.

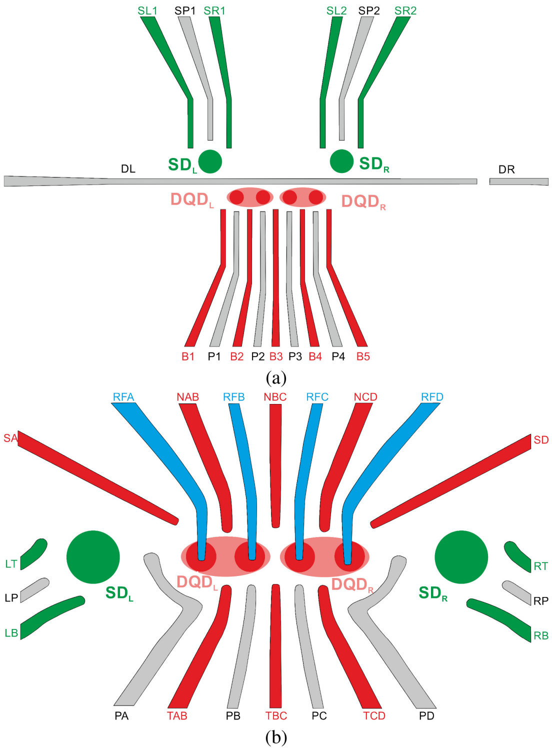


FIGURE 1. **(a) Gate layout 1 and (b) gate layout 2 for two double quantum dots (red dots) and two sensing dots (green dots). Barrier gates are marked in red, plunger gates in gray, sensor dot barrier gates in green, and radio frequency gates in blue.**

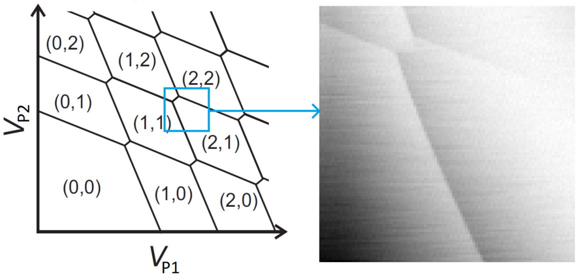


FIGURE 2. **Charge stability diagram [5, Ch. 8] and a measured CSD example of a double quantum dot.**

A. TUNING

The operation of gate-defined quantum dots requires a preceding tuning process to identify the gate voltages needed to capture and tunnel-couple individual electrons. The tuning consists of a coarse- and a fine-tuning part, comprising several subtasks themselves. One major subtask of the coarse-tuning is to form the quantum dots, represented by the four red dots in Fig. 1, and to deplete them to the single-electron regime. For this, one can observe changes in the dots' occupancy in CSDs, which reflect the charge as a function of the two plunger gate voltages. For example, the left part of Fig. 2 shows a modeled charge stability diagram (CSD) of a double dot with (l, r) denoting the number of electrons in the left and right quantum dot. The lines indicate an electron tunneling, and the honeycomb-like structure originates from the electrostatic coupling between the quantum dots and the distant gate [5, Ch. 8]. In the experiment, the line pattern appears when measuring the conductance (right part in Fig. 2) through the nearby electrostatically coupled sensing dot (green dots in Fig. 1). Unfortunately, the transition lines and occupation areas are often diffuse, disturbed by noise and artifacts, and contain only a smaller section of the whole CSD. Sometimes the transition lines are not even evident.

Generally, the CSDs contain relatively simple structures, but often with very low contrasts. Depending on the measurement procedure, the lighting is often nonuniform due to sensor dot drifts. Furthermore, with scalability of a quantum-dot-based platform for quantum computing in mind, the resolution should be as low as possible to minimize the measurement time. Conversely, natural images have a richer local covariance resulting from more complicated geometries. Their contrast ranges from medium to high with often uniform lighting and high image resolution [10]. For edge-preserving filters, especially, the low contrasts of CSDs are challenging, whereas their simple geometries ease structure preservation.

B. MOTIVATION

The most commonly used manual tuning procedure is time-consuming and not practicable for a scalable quantum computer with thousands of qubits. Therefore, several authors suggested solutions for an automated tuning procedure.

In coarse-tuning, machine learning techniques play a dominant role in determining the charge state of quantum dots. Frequently, we use convolutional neural networks (CNNs), which are artificial neural networks designed for image classification and pattern recognition [11]–[14]. [11] and [15] compared binary classifiers, such as logistic regression, multilayer perceptrons, support vector machines (SVMs), decision trees, random forests, quadratic discriminant analysis, and k-nearest neighbors (kNN) for determining if a CSD belongs to a single or double quantum dot. Moreover, Czischek *et al.* [16] established feed-forward neural networks (FFNNs). The methods above work with simulated datasets based on a physical model [12], [14], [16] or experimental data [13], [15] for training. Darulova *et al.* [11] compared synthetic and experimental training data results. They showed an advantage for experimental or a combination of synthetic and experimental data because the noise in experimental data was not understood well enough to simulate it accurately for synthetic training data.

Besides the supervised machine learning techniques, Lapointe-Major *et al.* [17] used for single quantum dot tuning signal preprocessing to remove the background of the CSDs and a modified Hough transform [18] or EDLines [19] to detect curved lines. Furthermore, Moon *et al.* [20] described an iterative unsupervised learning technique. Nevertheless, many approaches tune only a few device parameters, still require human intervention, or need nearly noiseless data.

Detecting the weak charge transition lines is a challenging task for a computer and even for a human expert if the quality of the diagrams is poor. Using preprocessing techniques that reduce unwanted noise while preserving or even improving the lines is beneficial for the quality of the results of the subsequent algorithms, including edge detection, segmentation, and pattern recognition. Hence, we examine the applicability and performance of edge-preserving smoothing filters on CSDs. Section I-C gives an overview of this class of filters.

The major aim is to find the transition lines in the CSDs. Knowing the position of the lead and interdot transitions, allows one to automatically determine additional quantum characteristics of the system by extracting parts (e.g., 1-D slices) from the original, nonfiltered data at corresponding positions. Primarily, these comprise different tunnel couplings by extracting the widths of the lead or interdot transitions [21]. However, when improving the structures, the filtered CSDs ease the identification of the gates' cross coupling from the slopes of the lead transitions for the generation of a virtual gate matrix [22].

Due to the different possible acquisition methods (transport or RF measurement, dc, or high-frequency voltage sources), the CSDs may contain different noise components and anisotropic resolutions, although isotropic scans are preferred. If the resolution is too anisotropic, even for resampled CSDs stepping effects might influence the filters' quality and subsequent methods for the automated tuning badly. The typical image size for our application is 100 × 100 pixels. This resolution is small enough to be acquired quickly and high enough to resolve transition lines. Overall, the selected CSD measurements comprise float values from the interval $[-0.3463, 0.6195]$ with an individual dynamic range between 0.0085 and 0.1696.

C. RELATED WORK

Edge-preserving smoothing filters address a base problem in signal and image processing to separate noise from the information. This section summarizes the most established methods, but the number of proposed algorithms is by far higher, often focusing on particular aspects and applications.

Local filters try to suppress the noise based on the information available in the pixel's neighborhood. The algorithm of Kuwahara *et al.* [23] defines four neighborhood regions and computes the mean of the region with the minimum variance. Working similarly, Nagao and Matsuyama [24] used nine rotating bars instead. Davies and Rosenfeld [25] took the mean of k-nearest neighbors from an ordered list, whereas Jee [26] worked with local statistics, an estimate for the global noise variance, and the minimum mean-squared error (MMSE). Another approach, the modified trimmed mean [27], combines the median and mean filters to exclude strongly divergent neighbors from averaging. Furthermore, the SUSAN technique [28] analyzes masks indicating the similarity between the local radius region and the center pixel. Besides, Paris *et al.* [29] suggested a pyramidal local Laplacian filter approach.

Some local filters work adaptively and are often used to remove speckle noise. Frost *et al.* [30] worked with an exponentially weighted convolution kernel that adapts to regions by exploiting local statistics. Similarly works the sigma filter [31], which computes the weighted average of pixels, whose intensities lie within the central pixel's two standard deviation range. Instead, the adaptive Wiener filter [32] uses weights derived from a minimum estimator and local estimated statistics. In addition, the gamma-maximum a posteriori (GMAP) filter [33] approximates the signal and noise statistics using the gamma distribution function and computes the denoised image using the MAP estimation. The iterative approach of [34] convolves the image with an averaging mask, depending on the signal continuity. However, the LLSURE filter [35] uses an affine transform on blocks, whose transform coefficients are determined by minimizing Stein's unbiased risk estimate (SURE).

Prevalent are methods that consider smoothing as a diffusion process. In the pioneering work of [36], the local image gradient influences the diffusion coefficients of the anisotropic heat diffusion equation. Black *et al.* [37] proposed a robust statistical interpretation of anisotropic diffusion (AD) that improves edge continuity. Moreover, Weickert [38] introduced semiimplicit schemes using additive operator splitting to reduce the number of iterations, whereas the nonlinear Gaussian filter chain (NLGC) [39] focuses on solving the partial differential equation by introducing a fixed changing rule for the Gaussian range and spatial distance kernels.

The bilateral filter (BF) [40] uses the same kernels. It computes the average of neighboring pixels weighted by the Gaussian of their range and spatial distance. However, the brute-force implementation has a high runtime complexity. Thus, several approaches aim to reduce runtime [41]–[44], often with lower accuracy and more artifacts. Besides acceleration, Zhang and Gunturk [45] studied the optimal BF parameter selection with their multiresolution BF approach.

The trilateral filter (TF) [46] extends the BF by tilting the filter window by the local image gradient, enabling piecewise linear smoothing. Another TF variant [47] removes mixed Gaussian and impulse noise using the local rank-ordered absolute difference (ROAD) statistic. Liu *et al.* [48] showed that the latter two TFs can be combined to increase noise reduction.

The guided image filter (GF) [49] uses structural information of a guidance image to estimate the parameters of a spatially variant filter kernel through linear regression. Furthermore, the fast guided filter [50] speeds up GF by sub- and upsampling techniques, whereas the weighted GF [51] improves quality by introducing edge-sensitive weights to the regularization parameter. Finally, the rolling guidance filter (RGF) [52] establishes an iterative and fast converging framework to control detail smoothing under a scale measure.

Besides the local filtering approaches, global methods also are used. The nonlocal means filter (NLM) computes a weighted mean of all pixels, taking the pixel's similarity to the target pixel into account. Mahmoudi and Sapiro[53] identified dissimilar neighborhoods and excluded them from the weighted averaging to accelerate NLM. Moreover, the unsupervised information-theoretic adaptive image filtering (UINTA) [54] realizes an iterative NLM method that improves the predictability of pixel intensities by minimizing the joint entropy. However, Buades *et al.* [55] determined the means' weights by comparing image patches. The optimal spatial adaption (OSA) method [56] performs denoising in adaptive patches using block-based weights, whereas Elad and Aharon [57] used K-clustering with singular value decomposition (SVD) to approximate patches as a linear combination of patches from a redundant dictionary.

In a different global approach, Rudin *et al.* [58] minimized the estimated image's total variation (TV) norm using a Lagrange multiplier and the gradient projection method, whereas Osher *et al.* [59] proposed an iterative Bregman regularization procedure for TV. Furthermore, Gilboa and Osher [60] introduced a nonlocal TV using the nonlocal gradient as the regularization term.

As a nonlinear global approach, [61] suggested a weighted least squares (WLS) optimization framework that involves the similarity between input and output and the weighted L2 norm penalty on image gradients. In a different method, the L0 gradient minimization optimizes an energy function based on prior gradient sparsity [62]. For acceleration, the fast WLS method [63] decomposes the original large linear system into two spatial dimensions and solves the matrix with a sequence of 1-D fast solvers.

By far, most transform-based methods use the wavelet transform. One of the first methods (VisuShrink [64]) uses standard wavelets and performs coefficient hard thresholding. Its soft-thresholding variant [65] shrinks the high-magnitude wavelet coefficients with a nonlinear function above the threshold. SureShrink [66] applies thresholding by minimizing the mse that takes a hybrid of the universal and the SURE threshold. Instead of the mse, BayesShrink [67] uses the Bayes risk estimator function. Furthermore, BiShrink [68] incorporates a bivariate probability density function to describe the statistical dependence between the wavelet coefficients of two adjacent scales. Another approach (NeighShrink [69]) considers deterministic neighborhood dependencies of coefficients, whereas the method in [70] minimizes the generalized cross-validation function. Finally, Silva *et al.* [71] applied edge strength-based adaptive thresholding to the wavelet decomposition of image blocks.

The wavelet transform usage also has been reported in combination with other methods. Dabov *et al.* [72] presented the block-matching and 3-D-transform-domain filter (BM3D) that combines the NLMs theory and wavelet transform-based denoising. Luisier *et al.* [73] proposed an algorithm that combines wavelets with SURE-LET, and Fattal [74] constructed adaptive wavelets using a robust data prediction lifting scheme for better data decorrelation. The method in [75] derives a hidden graph structure from the wavelet coefficients to model the image's prior probability.

Our goal is to identify methods providing robust and good results for CSD denoising. Thus, we preferably select well-established filters with available implementation and aim to cover a broad range of methodologies. We describe the chosen filters and their parameters in the following section in more detail.

II. FUNCTIONALITY AND PARAMETERS OF THE SELECTED FILTERS

In the following, $I(\mathbf{a})$ denotes the image value of a pixel at the 2-D space coordinate $\mathbf{a}$. We describe the parameters' optimization for the different approaches in Section IV.

A. BILATERAL FILTER

The bilateral filter (BF) [40] works with a weighted mean that is computed according to (1), where $N(\mathbf{a})$ is a normalization value.

$$\hat{I}(\mathbf{a}) = \frac{1}{N(\mathbf{a})} \sum_{\mathbf{b}\in D} I(\mathbf{b}) \cdot w_d(\|\mathbf{b}-\mathbf{a}\|) \cdot w_r(I(\mathbf{b}) - I(\mathbf{a})) \tag{1}$$

The domain function w_d weights closer pixels stronger, and the range function w_r similar pixels. Both weighting functions are typically Gaussians

$$w(\|x\|) = \exp\left(-\frac{\|x\|^2}{2\sigma^2}\right)$$

with different standard deviations, σ_d and σ_r, as the filter's parameters.

We use the filter implementation from DIPimage [76].

B. ANISOTROPIC DIFFUSION

The AD filter, published by Perona and Malik [36], implements an iterative process according to

$$\begin{aligned} I_{i,j}^{t+1} &= I_{i,j}^{t} + \lambda[c_N \cdot \Delta_N I + c_S \cdot \Delta_S I \\ &\quad + c_E \cdot \Delta_E I + c_W \cdot \Delta_W I]_{i,j}^{t}. \end{aligned} \tag{2}$$

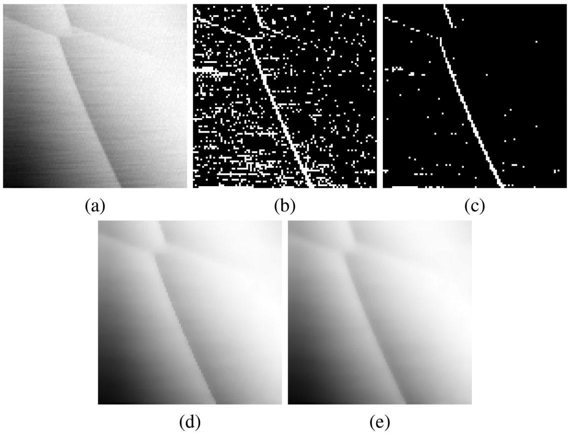
In this equation, t denotes the iteration step, $\Delta_k I_{i,j}$ the finite difference to the neighbor in k-direction, c_k the conduction coefficient in k-direction, and $\lambda \in [0, 1]$ the strength of the picture changes in each iteration. The conduction coefficients change in every iteration, exemplarily shown in the following equation for the northern direction:

$$c_{N_{i,j}}^{t} = g(\|(\nabla I)_{i+(1/2),j}^{t}\|) \approx g(|\Delta_N I_{i,j}^{t}|). \tag{3}$$

The function g is a nonnegative monotonic decreasing function with $g(0) = 1$. For our investigation, we use

$$g(|\Delta_\eta I|) = \exp\left(-\frac{|\Delta_\eta I|}{K}\right)$$

where Δ_η denotes the finite difference in direction η. Equation (3) smoothens the gradient magnitudes below the gradient threshold K and enhances the ones above. Perona and Malik suggested setting K to the 90% quantile of the gradient magnitude histogram, as described by Canny [77]. Fig. 3(c) shows the gradient magnitude thresholded with the 90% quantile for the example CSD in Fig. 3(a). The rightmost edge in the image disappears in the thresholded gradient magnitude. If we set K to the 70% quantile, more edge pixels


(a) (b) (c)

(d) (e)

FIGURE 3. (a) Example CSD, (b) its gradient magnitude thresholded with the 70% and (c) 90% quantile, and (d) the AD's result with $K = 70\%$ quantile and (e) $K = 90\%$ quantile.

are present in thresholded gradient magnitude, leading to a less blurred edge. On the other side, Fig. 3(b) also contains much noise, but this only has a negligible effect on the noise reduction visible in Fig. 3(d) and (e). These images also show that the lower value for K leads to sharper edges.

We use the anisotropic diffusion filter (AD) implementation from DIPimage [76] in our work.

C. NONLINEAR GAUSSIAN FILTER CHAIN

The NLGC [39] comprises iterated BFs with Gaussian weight functions. First, it reduces the contrast of fine details in the image while keeping the coarser structures which then are sharpened by gradually decreasing σ_r and increasing σ_d. The whole filter process can be represented by

$$GC = G_{\sigma_{r_k},\sigma_{d_k}} \circ G_{\sigma_{r_{k-1}},\sigma_{d_{k-1}}} \circ \cdots \circ G_{\sigma_{r_1},\sigma_{d_1}}(I)$$

$$\sigma_{r_{i+1}} = \frac{1}{\alpha} \cdot \sigma_{r_i}, \ \ \sigma_{d_{i+1}} = \sqrt[n]{\alpha^2} \cdot \sigma_{d_i},$$

where $G_{\sigma_{r_i},\sigma_{d_i}}$ is a BF with parameters σ_{r_i} and σ_{d_i}, k is the number of iterations of the filter chain, and n is the dimension of the image [78].

Aurich and Weule [39] observed that one could get best results for a two-dimensional image when halving σ_r and doubling σ_d after each step, i.e., $\alpha = 2$. Moreover, Mühlhaus [78] described a strategy for the initial determination of the parameters

$$\sigma_{r_1} = 2 \cdot \sigma \text{ and } \sigma_{d_1} = \frac{\sqrt[n]{4\alpha^2}}{2\sqrt{\pi}}$$

where σ is the standard deviation of noise in the image. Although one can determine σ via noise estimators, their precision for our data is unknown, causing us to treat σ as a free parameter. Additionally, Mühlhaus suggested using the chain length

$$k = \left\lceil \log_2 \left(\frac{H_{\min}}{2\sigma} \right) \right\rceil$$

where $H_{\min}$ is the minimum height of a preserved edge. We include only $H_{\min}$ and σ in the parameter optimization process and implemented the nonlinear Gaussian filter chain (NLGC) ourselves.

D. UNIVERSAL TRILATERAL FILTER WITH IMPULSE DETECTOR

The universal TF with impulse detector by Liu [48] comprises the two different TFs by Choudhury and Tumblin [46] and by Garnett *et al.* [47].

In addition to the BF, the TF tilts the filter window by a smoothed version of the image gradient. For this, Liu used the TF by Garnett, which introduces a third weighting function

$$w_i(\mathbf{a}) = \exp\left(-\frac{\text{ROAD}(\mathbf{a})^2}{2\sigma_i^2}\right) \tag{4}$$

to suppress impulse noise. This function uses ROAD that correspond to the sum of the four lowest gray value differences between a pixel and its eight neighbors. Moreover, a switching function

$$J(\mathbf{a}, \mathbf{b}) = 1 - \exp\left(-\left(\frac{\text{ROAD}(\mathbf{a}) + \text{ROAD}(\mathbf{b})}{2}\right)^2 \Big/ (2\sigma_j^2)\right) \tag{5}$$

is used that ensures for outliers the usage of w_i or w_r, otherwise. Then, the smoothed gradient

$$G_\theta(\mathbf{a}) = \frac{1}{N(\mathbf{a})} \sum_{\mathbf{b} \in D} \nabla I(\mathbf{b}) \cdot w_d(\|\mathbf{b} - \mathbf{a}\|) \cdot w_r(\nabla I(\mathbf{b}) - \nabla I(\mathbf{a}))^{1-J(\mathbf{a},\mathbf{b})} \cdot w_i(\mathbf{b})^{J(\mathbf{a},\mathbf{b})} \tag{6}$$

of the image can be calculated, where $N(\mathbf{a})$ normalizes the filter weights.

Finally, the main filter process is given by

$$\hat{I}(\mathbf{a}) = I(\mathbf{a}) + \frac{1}{N_\Delta(\mathbf{a})} \sum_{\mathbf{b} \in D} I_{\Delta,\mathbf{a}}(\mathbf{b}) \cdot w_d(\|\mathbf{b} - \mathbf{a}\|) \cdot w_r(I_{\Delta,\mathbf{a}}(\mathbf{b})) f_{\theta,\mathbf{a}}(\mathbf{b}) \tag{7}$$

$$\text{with } I_{\Delta,\mathbf{a}}(\mathbf{b}) = I(\mathbf{b}) - P_{\mathbf{a}}(\mathbf{b})$$

$$\text{and } P_{\mathbf{a}}(\mathbf{b}) = I(\mathbf{a}) + \langle G_\theta(\mathbf{a}), \mathbf{b} - \mathbf{a} \rangle.$$

$\hat{I}(\mathbf{a})$ denotes the value for the filtered pixel, $N_\Delta(\mathbf{a})$ is a normalization factor, and the function $f_{\theta,\mathbf{a}}$ represents together with a threshold R a binary decision for gradient similarity between $\mathbf{a}$ and $\mathbf{b}$.

The universal TF contains many parameters that need to be determined. Garnett *et al.* suggested setting σ_r in (6) to twice the standard deviation of the Gaussian noise in the image, estimated by a modified version of [79]. Additionally, Choudhury and Tumblin proposed to set the threshold R in (7) to σ_r. Moreover, Garnett *et al.* showed that the ROAD values for uncorrupted pixels are likely to lie below 23.4% of the data range, so we used this value for σ_i in (4) and

(6), respectively. Moreover, the switching function (5) should also indicate if only one pixel (**a** or **b**) is an outlier, but it contains a mean operation. Therefore, we choose $\sigma_j = \sigma_i/2$ to avoid classifying different pixels as outliers by w_i and J.

Finally, σ_d, which can be equal for (6) and (7), and σ_r in (7) remain to be determined.

We based our code of the universal TF on the implementation of Choudhury and Tumblin's TF developed by Astola [80].

E. NONLOCAL MEANS

The NLM of Buades *et al.* [55] computes a weighted average based on the analysis of neighborhoods instead of pixels to achieve a more robust comparison. It computes the new pixel value

$$\hat{I}(\mathbf{a}) = \frac{1}{N(\mathbf{a})} \sum\nolimits_{\mathbf{b}\in\mathcal{W}} I(\mathbf{b}) w(\mathbf{a},\mathbf{b}) \tag{8}$$

with no spatial distance playing a role in the weights anymore. In (8), $N(\mathbf{a})$ is a normalization value, and $\mathcal{W}$ is the window for the search for similar neighborhoods. The weight function

$$w(\mathbf{a},\mathbf{b}) = \exp\left(-\frac{\|I(\mathcal{N}_{\mathbf{a}}) - I(\mathcal{N}_{\mathbf{b}})\|^2_{2,\sigma}}{h^2}\right)$$

depends on the similarity of the neighborhood regions $I(\mathcal{N}_{\mathbf{a}})$ and $I(\mathcal{N}_{\mathbf{b}})$ around **a** and **b**. The numerator in the weight function represents a Gaussian weighted Euclidean distance with standard deviation σ. Furthermore, high values for the filtering degree h cause stronger blurred filter results. Different from the suggestion of Buades *et al.*, we set $\mathcal{W} = 3$ and $\mathcal{N} = 9$ as larger values do not create better results but only generate longer runtimes.

In our work, we use the implementation of Manjon-Herrera [81]. Instead of the Gaussian kernel, it uses weights depending on the size of $\mathcal{N}$.

F. GUIDED IMAGE FILTER

He *et al.* [49] published the GF. Besides the input image, they used a guidance image to determine the weights for the weighted average. When using the input image itself as the guidance image, the filter acts as an edge-preserving smoothing filter

$$\hat{I}(\mathbf{a}) = \sum_{\mathbf{b}\in D} W_{\mathbf{a},\mathbf{b}}(I) I(\mathbf{b}) \text{ with}$$

$$W_{\mathbf{a},\mathbf{b}}(I) = \frac{1}{|\omega|^2} \sum_{k:(\mathbf{a},\mathbf{b})\in\omega_k} \left(1 + \frac{(I(\mathbf{a}) + \mu_k)(I(\mathbf{b}) + \mu_k)}{\sigma_k^2 + \epsilon}\right).$$

The variable ω_k corresponds to the filter window, which contains pixels **a** and **b**, $|\omega|$ denotes the size of the filter window, μ_k and σ_k^2 are the mean and variance in ω_k, and ϵ is a regularization parameter.

The window sizes $|\omega|$ and ϵ are the parameters to be determined. They have a similar impact as the parameters σ_d and σ_r in the BF [49].

We use the GF implementation of Pan [82] for this work.

TABLE I Standard Deviations for Added Pink and White Gaussian Noise

	σ_p	σ_w
distinct edges	0.00025 (p1) 0.0005 (p2) 0.001 (p3)	0.000125 (w1) 0.00025 (w2) 0.0005 (w3) 0.001 (w4) 0.002 (w5)
indistinct edges	0.000125 (p0) 0.00025 (p1) 0.0005 (p2)	0.0000625 (w0) 0.000125 (w1) 0.00025 (w2) 0.0005 (w3) 0.001 (w4)

TABLE II Scales and Standard Deviations for Added RTN

	scales	σ_{rt}
distinct edges	[5,50]	0.001 (t1)
indistinct edges	[5,50]	0.0005 (t2)

TABLE III Optimal Parameter Ranges of the Filters for the Artificial and Experimental Datasets

Filter	Parameter	Artificial data	Experim. data
BF	σ_d σ_r $[\cdot 10^{-3}]$	[0.707, 5.000] [0.469, 14.307]	[0.672, 5.000] [1.140, 13.654]
TF	σ_d σ_r $[\cdot 10^{-3}]$	[1.000, 5.000] [0.285, 50.000]	[0.622, 5.000] [1.755, 50.000]
AD	λ k $[\cdot 10^{-3}]$ iterations	[0.129, 0.478] [0.667, 11.532] [1, 49]	[0.122, 0.537] [1.726, 12.359] [1, 49]
TV	λ iterations	[0.056, 1.000] [11, 199]	[0.150, 1.000] [43, 199]
NLGC	σ $[\cdot 10^{-3}]$ H_{min} $[\cdot 10^{-3}]$	[0.147, 9.716] [0.032, 10.313]	[0.560, 7.406] [0.463, 73.224]
GF	ϵ $[\cdot 10^{-3}]$ filter size	[0.000, 1.000] [1, 2.5]	[0.002, 0.775] 1
NLM	σ $[\cdot 10^{-3}]$	[0.155, 3.000]	[0.561, 5.361]
RGF	σ_d σ_r $[\cdot 10^{-3}]$	[0.750, 3.000] [0.364, 6.263]	[0.631, 2.989] [0.591, 11.592]
BM3D	σ $[\cdot 10^{-3}]$	[0.274, 7.359]	[0.558, 4.521]
SS	σ $[\cdot 10^{-3}]$ no. levels wavelet	[0.230, 4.222] [3, 4] bior 2.2	[0.975, 2.918] [3, 4] bior 2.2

G. ROLLING GUIDANCE FILTER

The RGF by Zhang *et al.* [52] constitutes an iterated version of the joint BF [83] and uses a guidance image to compute the range weights w_r. In the beginning, the guidance image is constant. For the next iterations, the guidance image originates from the result of the previous iteration. The iterative filtering

$$J_{t+1}(\mathbf{a}) = \frac{1}{N(\mathbf{a})} \sum_{\mathbf{b}\in D} I(\mathbf{b}) \cdot w_d(\|\mathbf{b} - \mathbf{a}\|)$$
$$\cdot\, w_r(J_t(\mathbf{b}) - J_t(\mathbf{a}))$$

comprises a normalization value $N(\mathbf{a})$ and weighting functions w_d and w_r such as in the BF.

The free parameters are the number of iterations and σ_d and σ_r of the joint BF. However, Zhang *et al.* stated that the pixel values only change slightly after ten iterations. Thus,

TABLE IV Runtime and Parameter Determination for Different Filters

	Mean Runtime	Parameter determination
BM3D	0	++
NLM	--	+
BF	+	-
NLGC	0	-
AD	-	-
RGF	0	-
TF	-	-
GF	+	0
TV	0	-
SS	-	+

We rate the mean runtime from up to 0.1 S (+), up to 0.4 S (0), up to 2 S (-), and up to 12 S (–). The rating for parameter determination ranges from no parameter (++), to only one parameter (+), to one variable and one constant parameter in our evaluation (o), and to two or more parameters (-).

we fix the number of iterations to ten and use their implementation [84] for further investigations.

H. TOTAL VARIATION DENOISING

The TV denoising filter, developed by Rudin *et al.* [58], minimizes a cost function. In the optimization problem

$$\underset{\hat{I}\in \mathrm{BV}(\Omega)}{\arg\min} \|\hat{I}\|_{\mathrm{TV}(\Omega)} + \frac{\mu}{2}\int_{\Omega}(\hat{I}(\mathbf{a}) - I(\mathbf{a}))^2 d\mathbf{a}$$

the first term denotes the TV of the smoothed image $\hat{I}$, indicated by the TV seminorm $\|\cdot\|_{\mathrm{TV}(\Omega)}$ over the domain Ω. The second term ensures that the solution to the optimization problem is close to the observed noisy image I [85]. The solution's search space comprises all bounded variation images $\mathrm{BV}(\Omega)$, and the parameter μ affects the balance between removing the noise and preserving the signal content.

Using the L1 norm instead of the L2 norm in the second term results in a better outlier rejection [86]. One can solve this so-called TV-L1 problem with the primal-dual algorithm [87].

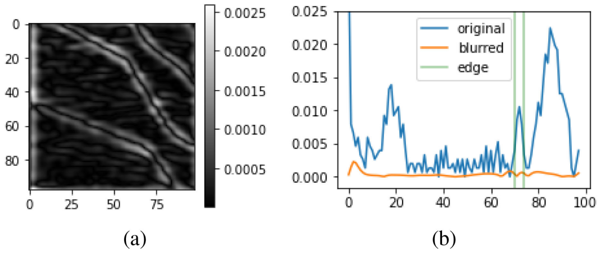
We use the TV-L1 implementation of Lourakis [88] that requires μ and the number of iterations for the primal-dual algorithm as input parameters.

I. BLOCK MATCHING AND 3-D FILTERING

With their BM3D, Dabov *et al.* [72], [89] combined the theory behind NLM and wavelet transform-based denoising.

The filter splits up into two steps. First, it computes a basic estimate for the denoised image used afterward, together with the noisy image, to generate a final estimate.

The first step in both estimation processes, called grouping, looks for similar image fragments (blocks) in the noisy image and groups them into 3-D data arrays (groups). Additionally, this step forms the second set of 3-D groups from the basic estimate for the final estimate.



(a) (b)

FIGURE 4. (a) Second-order derivative of a blurred (Gaussian filtered) CSD and (b) its 42nd row (orange) together with the second-order derivative of the original version (blue). In the manually marked edge region (green area), the second-order derivative effectively indicates the sharpness of the edge. The second-order derivative is not used for edge detection here.

The next step for the basic estimate is collaborative hard thresholding. It comprises a 3-D transformation, in our case separated into a 2-D transformation with the biorthogonal (bior) 1.5 wavelet and a 1-D transformation with the Haar wavelet, applied to the groups and subsequent hard thresholding of the transformation coefficients. Then, it computes an estimate of all grouped blocks by inverting the 3-D transformation. These steps are similar to the computation of the final estimate. However, collaborative Wiener filtering is applied here to the groups of the noisy image using the energy spectrum of the basic estimate.

Finally, the potentially overlapping blockwise estimates are aggregated to pixelwise estimates via a weighted average in both steps.

In this process, several parameters play a role. Nevertheless, Dabov *et al.* propose parameter sets for various use cases and noise intensities [72]. For our application, we use the standard profile in the implementation of Mäkinen *et al.* [90]. It only requires the standard deviation of the noise, which we assume to be unknown and, thus, feeds into the parameter optimization process.

J. SURESHRINK

The SureShrink filter (SS), developed by Donoho and Johnstone [66], assumes that most image information is present in the highest wavelet transform coefficients. Thus, it shrinks or even removes the lowest coefficients associated with noise. A threshold chosen to minimize the mean squared error (mse) between the original and noisy image steers this process. For this, the algorithm minimizes the stein unbiased risk estimate (SURE) [91].

The required parameters for this algorithm are the wavelet type, the number of levels for the wavelet decomposition, and the standard deviation of noise in the image, which we estimate as twice the noise estimated by Donoho's noise estimator [65]. For the evaluation of the SureShrink filter (SS), we select the implementation by Verma [92].

III. EVALUATION MEASURES FOR THE FILTER RESULTS

To evaluate the performance of the filters in our application, we use two approaches. The first approach works with

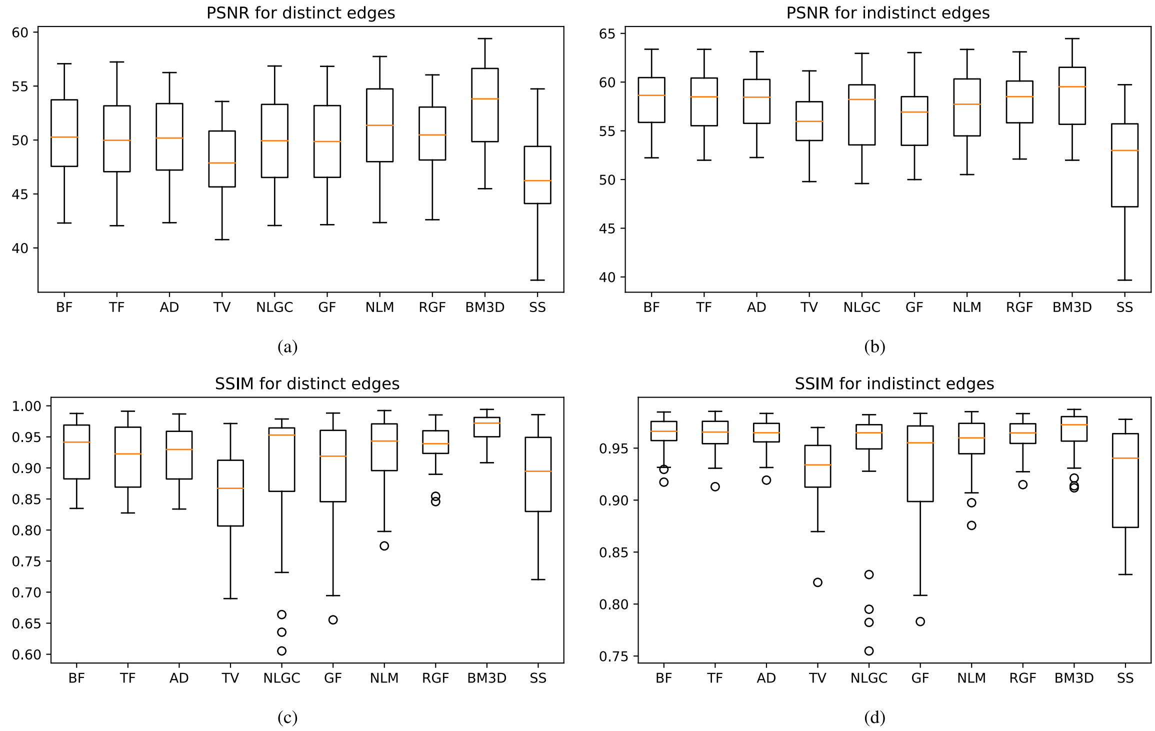


FIGURE 5. Boxplots of the filters' PSNR and SSIM results for artificial datasets with (a), (c) distinct and (b), (d) indistinct edges.

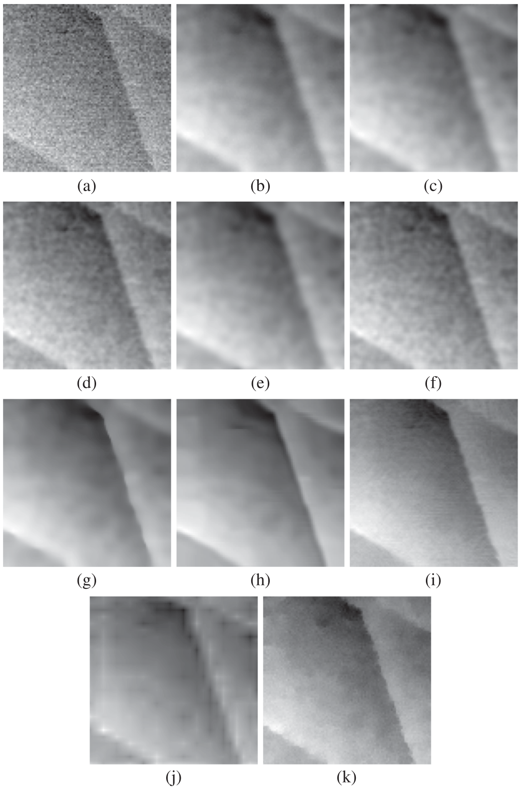


FIGURE 6. Results for artificial CSD with high-contrast edges and $\sigma_p = 0.001$ and $\sigma_w = 0.002$. (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

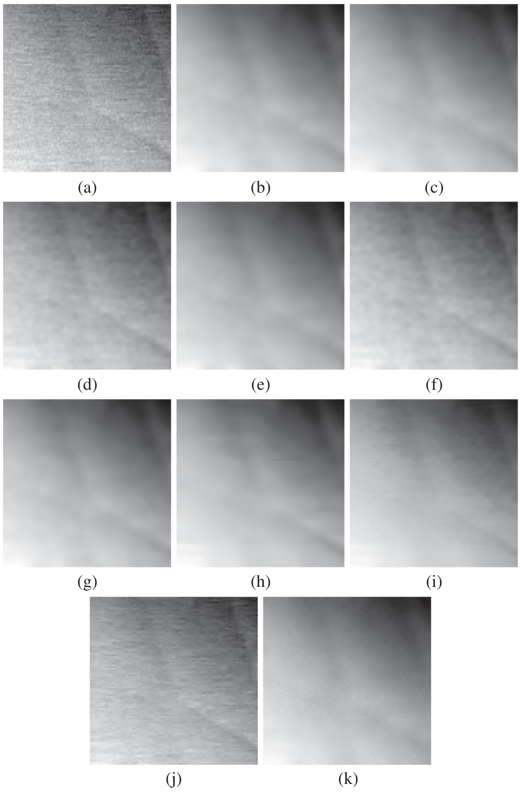


FIGURE 7. Results for an artificial CSD with low-contrast edges and $\sigma_p = 0.0005$ and $\sigma_w = 0.0005$. (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

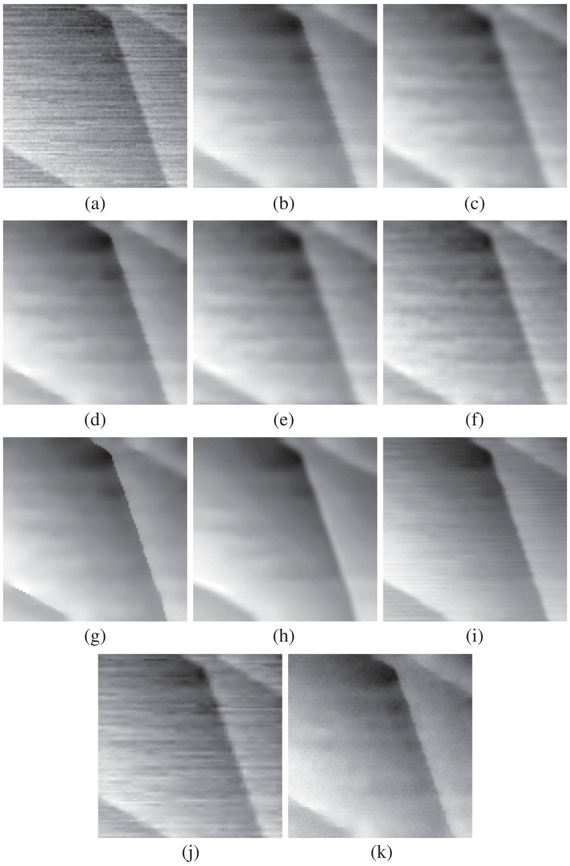


FIGURE 8. Results for an artificial CSD with high-contrast edges and $\sigma_p = 0.0005$, $\sigma_w = 0.0005$, and RTN. (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

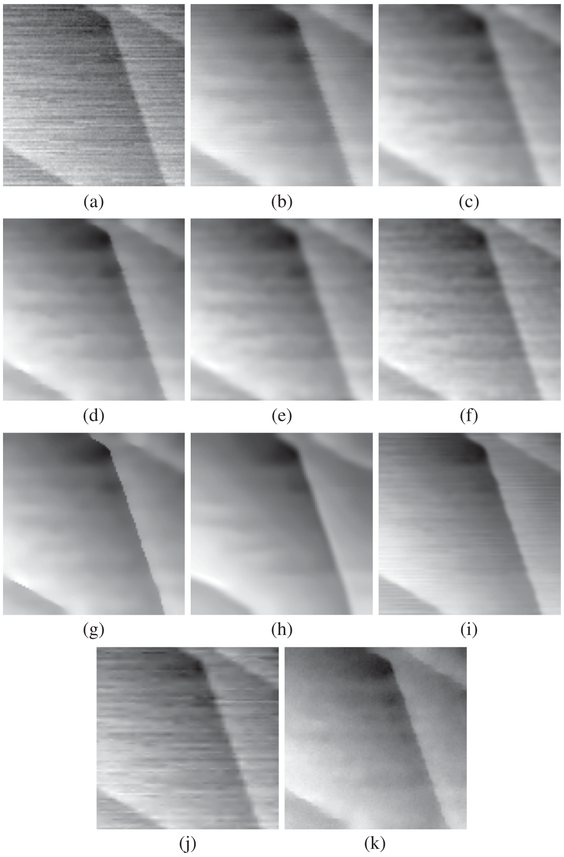


FIGURE 9. Results for an artificial CSD with low-contrast edges and $\sigma_p = 0.000125$, $\sigma_w = 0.000125$, and RTN. (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

generated artificial data that provide the evaluation's ground truth, allowing us to apply full-reference quality metrics. The second approach directly evaluates the filters' performance on the measured data via noise reduction and edge sharpness measures.

We choose the evaluation metrics based on the assumption that the filtered CSDs are used mainly for the detection of transition lines. Extracting the position of the lead transitions is the most important aspect here. Thereafter, one can derive the location of triple points and interdot transitions, which are often smoother. Lead transitions mainly smear out if the temperature is too high, but then the quantum dot system will not be operable anyway. Thus, the use of a sharpness measure constitutes a suitable evaluation metric.

This section explains the artificial data generation and introduces the used quality metrics.

A. GENERATION OF ARTIFICIAL DATA

First, we generate reference images by mimicking the characteristics of experimental data.[1] They include the typical honeycomb-like structure as well as linear progressions that are visible inside the different electron occupation regions and result from drifts in the sensor dot. Moreover, the size of the honeycombs changes with the occupation of the quantum dots, and the boundaries can be distinct or diffuse. Pronounced structures are especially beneficial to evaluate the edge preservation capability of the different filters.

Then, we add noise with different characteristics to the generated reference images. Darulova *et al.* [11] stated that the charge stability diagrams of GaAs double quantum dots contain six different noise types: white noise, $1/f$ noise, random telegraph noise (RTN), charge fluctuations on gates, low-frequency current modulations, and pinch-off current modulation. The latter two only affect the edges' sharpness that we already considered. Charge fluctuations due to defects in the semiconductor and latching due to too small measurement time or too small tunnel coupling result in a displacement of the transition lines in one of the CSD's axes. In our measured data, these artifacts do not play a role, but the algorithms under consideration will treat them as sharp edges without correcting the artifacts. This behavior is desired as these artifacts result from faulty hardware or bad measurement setup and, thus, should be detected before extracting the charge transitions in the CSDs. We add the remaining three,

[1] Ongoing work focuses on model-based simulation of CSDs with additional distortions that we plan to make publicly available.

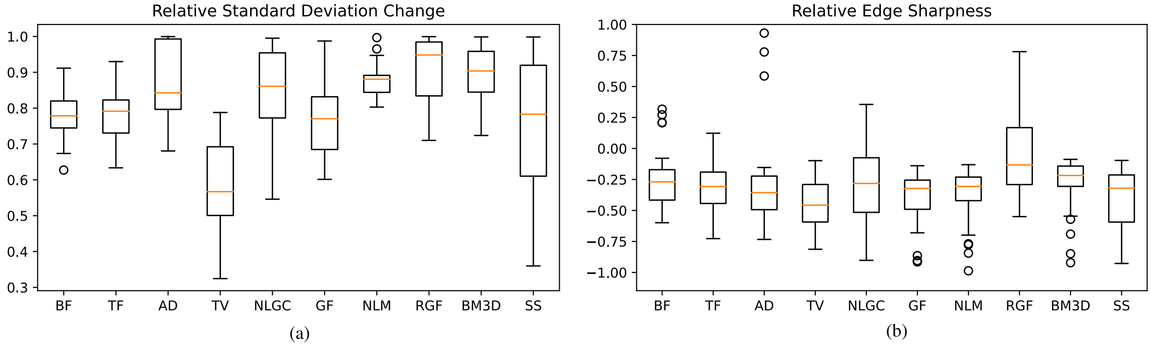


FIGURE 10. Boxplots of the filters' (a) RSC and (b) RES results. For a better resolution of the differences, some outliers were removed.

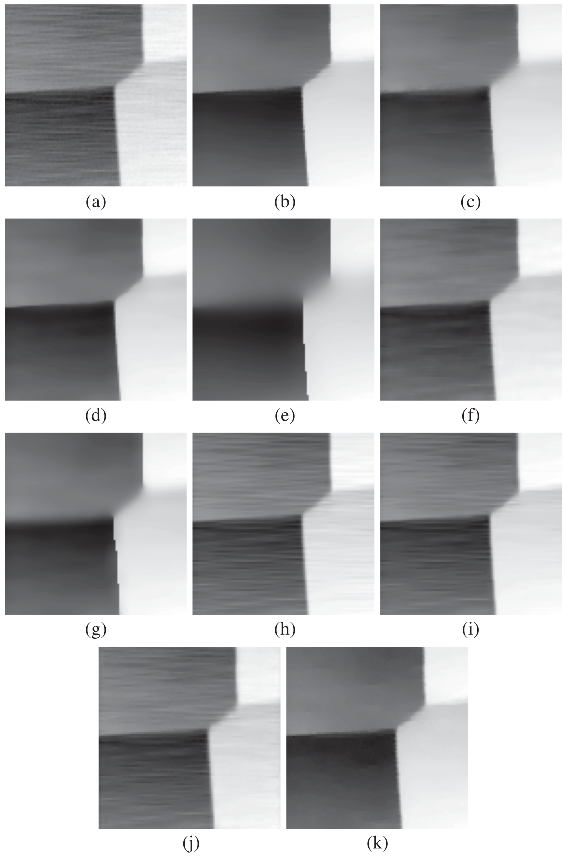


FIGURE 11. Visual results for a measured CSD (GL1; compensated gates) with high-contrast edges (dataset GL1-09 in Table V). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

RTN, white Gaussian, and 1/f Gaussian noise, with varying strengths to the artificial data.

Using the noise estimator by Immerkær [79], we calculate a guess for the total noise strength from the noise estimates' mean of experimental data. From this, we derive different standard deviations for the $1/f$ noise (σ_p) and choose the standard deviations for the white noise (σ_w) as 0.5, one, and two times σ_p (see Table I). In that way, we apply nine different noise combinations per picture.

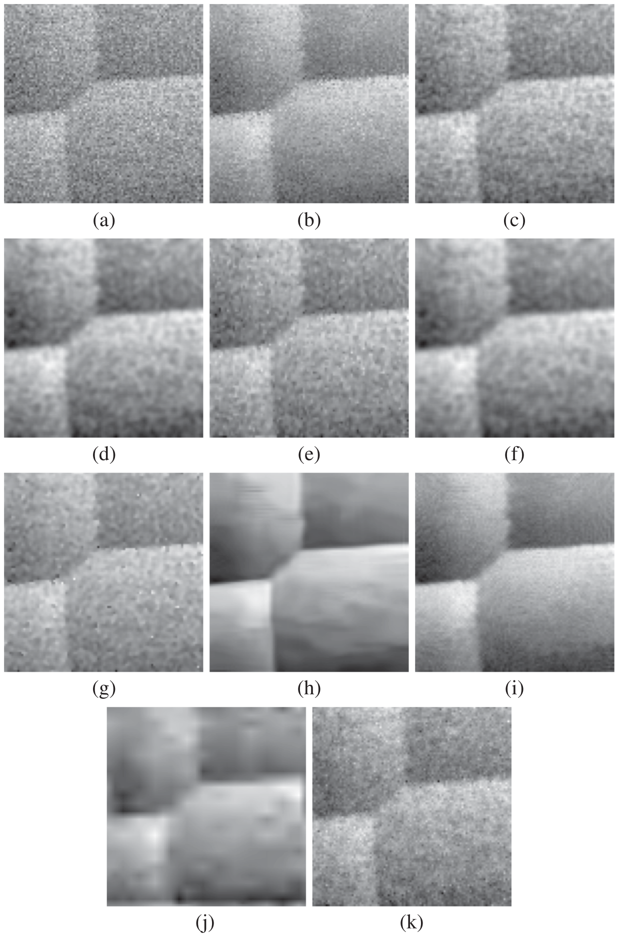


FIGURE 12. Visual results for a measured CSD (GL1; compensated gates) with intensive Gaussian noise (dataset GL1-12 in Table V). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

Even though specific algorithms might better remove RTN, it is of interest how the edge-preserving filters deal with it. Thus, we further create a dataset by adding RTN to the artificial data already containing white and $1/f$ noise. We use [93] for its generation with equally distributed scales and standard deviations (σ_{rt}), as depicted in Table II.

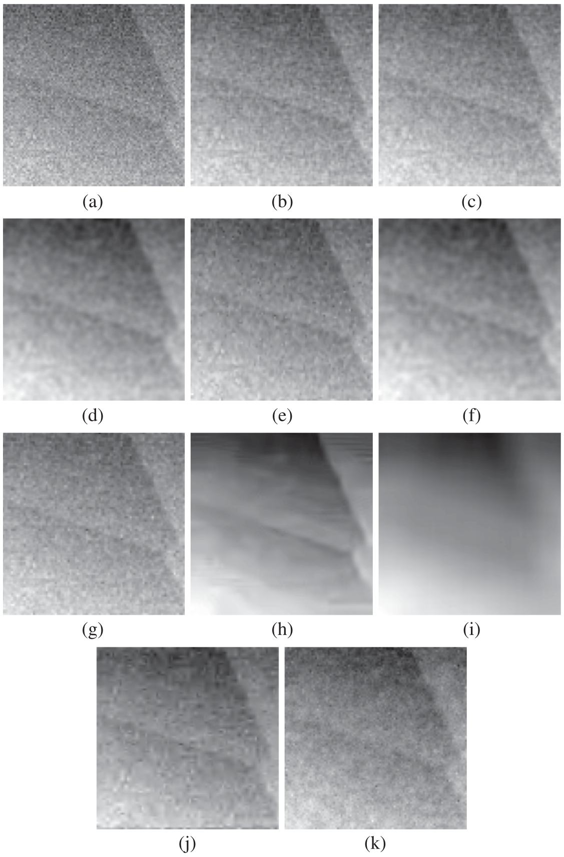


FIGURE 13. Visual results for a measured CSD (GL1; uncompensated gates) with low-contrast edges and intensive noise (dataset GL1-14 in Table V). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

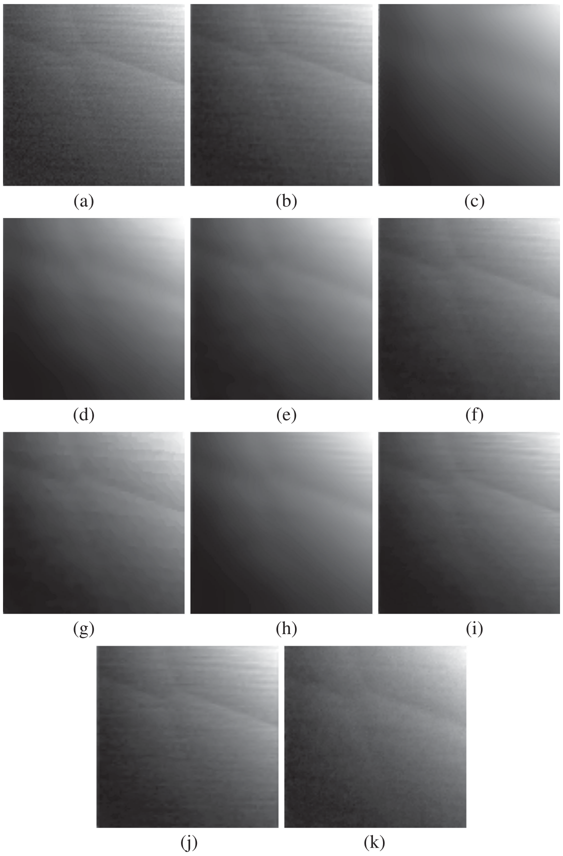


FIGURE 14. Visual results for a measured CSD (GL1; uncompensated gates) with low-contrast edges and low noise (dataset GL1-21 in Table V). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.

B. FULL-REFERENCE QUALITY METRICS

For the artificial datasets, we apply two full-reference quality metrics. The peak signal-to-noise ratio (PSNR) is computed by

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX}_I^2}{\mathrm{mse}}\right)$$

$$\text{with mse} = \frac{1}{N}\sum_{\mathbf{a}\in D}\left[I(\mathbf{a}) - \hat{I}(\mathbf{a})\right]^2$$

where N denotes the number of pixels, $\hat{I}$ the filtered image, and MAX_I the maximum pixel value observed in the images. It performs well on judging noise components [94], [95] but poorly reflects the visual quality. Thus, images with different distortions may have the same mse [96]. Therefore, we also apply the structural similarity index measure (SSIM) [97]. It primarily considers structural changes but also luminance and contrast masking terms.

C. QUALITY METRICS FOR MEASURED DATA

Optimizing the filter parameters for measured data requires a suitable metric. Unfortunately, due to missing ground truth, full-reference metrics are not applicable. Therefore, we introduce our own quality metrics for noise reduction and edge sharpness.

1) MEASURE FOR REDUCTION OF NOISE

We define the reduction of noise by the relative standard deviation change (RSC)

$$\mathrm{RSC} = \frac{\sigma_{\mathrm{noise}}(I) - \sigma_{\mathrm{noise}}(\hat{I})}{\sigma_{\mathrm{noise}}(I)}$$

where σ_{noise} represents the noise's standard deviation of the original image I or the filtered image $\hat{I}$, estimated with Immerkær's method [79].

2) MEASURE FOR EDGE SHARPNESS

Contrary to the first derivative, the second derivative's magnitude can significantly identify edge sharpness (see Fig. 4). Accordingly, our edge sharpness measure

$$\mathrm{ES}(I) = \overline{|(I \circ m_e) * \mathbf{D}^2|}$$

evaluates the mean of the second derivative via a Laplace filter $\mathbf{D}^2$ only at edge regions m_e we marked manually. This procedure allows us to safely define the edge regions and to

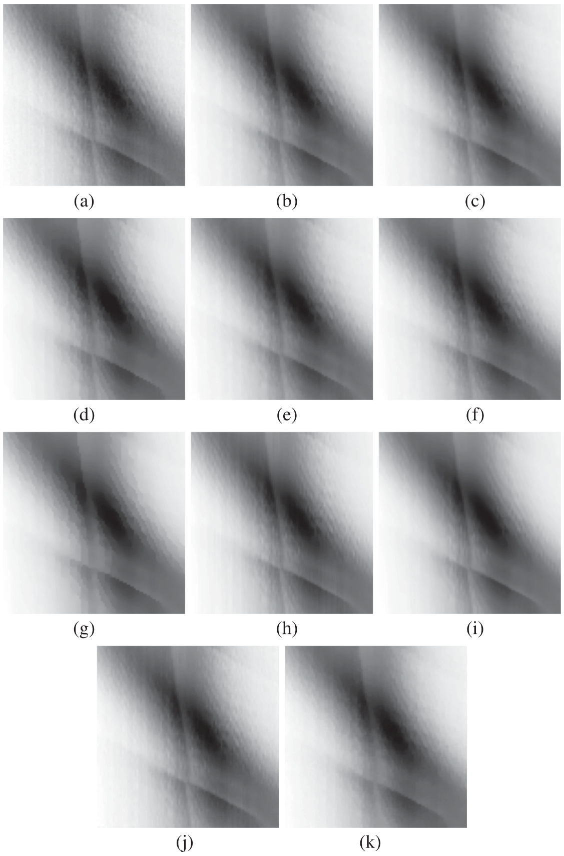


FIGURE 15. **Visual results for a measured CSD (GL2; uncompensated gates) with low-contrast edges and low noise (dataset GL2-08 in Table VI). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.**

improve the sensitivity of the metric. However, by additionally excluding the lower 10% of the second-derivative values from averaging, we disregard pixels that probably still belong to noise or background drifts.

Finally, the relative edge sharpness (RES)

$$\text{RES} = \frac{\text{ES}(\hat{I}) - \text{ES}(I)}{\text{ES}(I)}$$

compares the edge sharpness in the original image I and the filtered image $\hat{I}$. We only incorporate the preserved transitions into the computation to maintain comparability. Negative RES values indicate blurred edges; positive values indicate sharpened ones.

IV. EVALUATION OF THE FILTERING RESULTS

To ensure the best performance of each filter and, thus, comparability, we optimize the filters' parameters for every image individually, as described in Section IV-A. Sections IV-B and IV-C comprise the quantitative and qualitative evaluation of the filters' results by suitable metrics and visual assessment.

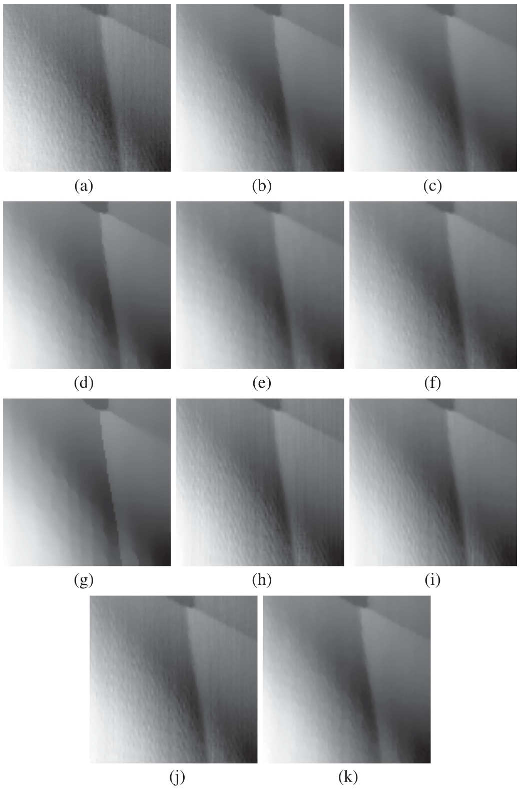


FIGURE 16. **Visual results for a measured CSD (GL2; uncompensated gates) with medium-contrast edges and low noise (dataset GL2-02 in Table VI). (a) Original. (b) BF. (c) TF. (d) NLGC. (e) AD. (f) GF. (g) RGF. (h) BM3D. (i) NLM. (j) SS. (k) TV.**

A. OPTIMIZATION OF FILTER PARAMETERS

In the optimization process, we use the PSNR for the artificial data and the sum of RSC and RES for the experimental data. As the incorporated Nelder–Mead algorithm [98] performs a minimum search, we adapt the quality metrics by negating them. For restricted parameters, such as λ of the AD, we use a bounded version of the Nelder–Mead algorithm [99]. In general, the SS performs best if using the bior 2.2 wavelet. The observed increased RES appearing with the discrete Meyer (dmey) wavelet results from pixelating artifacts.

Table III summarizes the optimal range for each parameter of the different filters.

B. EVALUATION FOR ARTIFICIAL DATA

We evaluate the filters' performance on the artificial datasets by metrics and visual appraisal. Fig. 5 shows the quality metric results for the datasets disturbed with pink and white Gaussian noise and Fig. 17 additionally interfered with RTN. Figs. 6–9 supplement these metrics by the filters' visual impact on exemplary CSDs with high- and low-contrast transitions.

1) PINK AND WHITE GAUSSIAN NOISE

For the high-contrast case [see Figs. 5(a) and (b) and 6], the BM3D performs best in terms of noise suppression and structure preservation metrics, whereas the TV and SS score the lowest quality results. Nevertheless, the other filters predominantly show promising results. However, the BF, TF, and AD visually blur edges more heavily, whereas the NLGC and GF maintain the noise in homogeneous areas but only slightly reduce the transition lines' visibility. The RGF accentuates high-contrast edges but lowers weak ones, as visible in the upper right corner of the CSD.

For the low-contrast case [see Figs. 5(b) and (d) and 7], the PSNR features an overall increased level as it tends to be less sensitive due to the lower high-frequency parts. Compared to the high-contrast case, especially the SSIM is less capable of revealing significant differences for the best filters. Nevertheless, the overall rating is similar. If intensive noise corrupts the CSDs, most filters visually remove it well but blur the edges heavily. In contrast, only the NLGC, GF, and SS exhibit shortcomings in noise removal but in return maintain edges relatively well.

2) ADDITIONAL RANDOM TELEGRAPH NOISE

The filters' quality metric results (see Fig. 17) with additional RTN differ from the results in Section IV-B1 mainly concerning the distributions' spread and mean's offset, but the assessment of the filters is similar. Thus, we refrain from discussing the values here again and only point out the filters' capability of suppressing RTN.

For datasets with high-contrast edges, the RGF, TV, and BM3D remove the RTN well (see Fig. 8). Only the BM3D leaves some single bursts in a few datasets. The SS, NLM, and BF keep much of the RTN and the NLGC leaves it visible near edges. The other filters treat the bursts like real edges and, thus, blur them similarly.

For indistinct edges, the NLGC, SS, and NLM cannot remove the RTN. The other filters considerably smudge the CSDs' contours but also leave smeared bursts visible sometimes.

C. EVALUATION OF EXPERIMENTAL DATA

We quantitatively study the quality of the different filters on the experimental data using the RES and RSC metrics. For visual appraisal, Figs. 11–16 show the filters' performance on characteristic CSDs.

1) METRIC RESULTS

Fig. 10 depicts the metrics' results for all selected experimental data. Therein, the RGF followed by the BM3D stand out due to their high capability of shaping distinct edges while effectively reducing noise. Furthermore, the NLGC, NLM, BF, and TF perform fair. This also holds, with small drawbacks, for the AD and GF, whereas the SS and TV underperform.

2) VISUAL APPRAISAL OF CSDS FROM GATE LAYOUT 1

The CSDs in Figs. 11–14 contain measurements using GL1 with (see Figs. 11 and 12) or without (see Figs. 13 and 14) virtual gates for cross talk and background disorder potential compensation [8]. The compensation results in axis-parallel transitions with improved contrast.

Not taking RTN into account, nearly all filters remove noise and preserve edges if compensated CSDs feature a high contrast (see Fig. 11). However, the AD, RGF, and TF tend to blur some transitions and the AD and RGF additionally produce staircase artifacts at some different borders. For compensated CSDs with intense Gaussian-like noise, the BM3D works remarkably fine, but the NLM also achieves a good result (see Fig. 12). All other filters cannot eliminate the noise effectively and often impair the edge sharpness.

Uncompensated CSDs from GL1 comprise low-contrast transitions and more decisive background disorder potential shifts that hinder the filters' ability to remove the noise.

However, the BM3D still eliminates intensive Gaussian-like noise fine and preserves the edges well (see Fig. 13). Although the NLM, which worked well for the compensated CSDs, erases the noise, it blurs the edges heavily. All the other filters cannot remove the strong noise sufficiently.

For uncompensated scans with low-level noise (see Fig. 14), the filters either blur the boundaries or do not effectively remove the noise. If any, only the BF, GF, RGF, and NLM slightly improve the CSDs. Overall, this reveals the challenging task for this kind of measured datasets.

3) VISUAL APPRAISAL OF CSDS FROM GATE LAYOUT 2

The CSDs for GL2 (see Figs. 15 and 16) contain considerable background disorder potential shifts, and the transition lines are less distinct.

Overall, the noise reduction for the low-contrast CSDs from GL2 is lower as it is more challenging to maintain the edges. There are only minor differences between the different filter results, and the improvement of the original datasets is poor. The NLGC and the RGF generate predominantly homogeneous backgrounds and very sharp edges, but the latter tends to staircase effects sometimes (see Fig. 15). All other filters also reduce noise but create slightly blurrier edges, avoiding staircase effects at the same time. However, the quality differences of the filters are less significant than for datasets from GL1.

4) RANDOM TELEGRAPH NOISE

RTN manifests itself in the measured or filtered CSDs as horizontal stripes (see Figs. 11 and 14). Dedicated supplementary algorithms can improve the quality of its removal. Being worth mentioning, some edge-preserving smoothing filters remove this type of noise in our data already well. Primarily, this is valid for the NLGC, RGF, and TV. Also, the BF, TF, AD, and GF usually remove the bursts successfully. However, a supplementary burst removal algorithm would be helpful, especially for images where the strengths of the edges and the RTN are at similar scales.

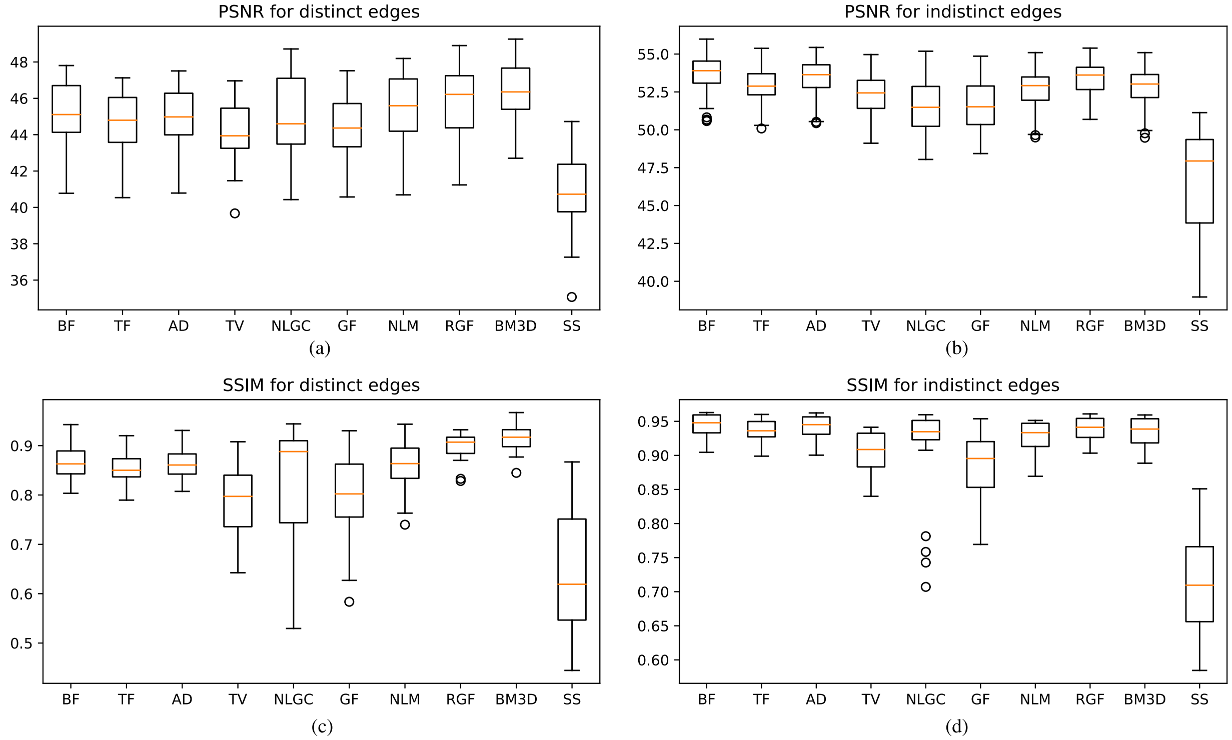


FIGURE 17. **PSNR and SSIM values for artificial datasets with distinct and indistinct edges and additional burst noise.**

The BM3D, NLM, and SS leave much RTN in the datasets. Although, for the BM3D, this might result from the parameter determination algorithm, this filter removes RTN well for the artificial data.

D. PARAMETER DETERMINATION AND RUNTIME BEHAVIOR

One important aspect of automated tuning is the determination of suitable filter parameters without human interaction. In this regard, the BM3D has the advantage that it only requires the standard deviation of the noise in the experimental data, which might be extracted by a noise estimator (here [79]). Usually, the measured noise reduction is slightly lower in that case, but the visual results are still very similar to those with optimal parameters. All the other filters require a strategy for parameter determination without ground truth.

Furthermore, the runtime of the filters is another critical aspect for the automated tuning of thousands or millions of qubits in the future. Being only one part of the transition detection, this preprocessing step should not be a limiting factor. Depending on the used method, typical acquisition times for a 100 × 100 CSD vary between 1 ms (AWG, RF readout) and 10 s (DAC, dc readout), which gives an indication for the overall computation time when parallelizing analysis and the next dataset acquisition. Even though the implementations and their optimizations are very different, we can give a runtime assessment based on measured values. We measured the runtimes for images with size 100 × 100 on a server with an Intel Xeon processor E5-2630 v2 with 2.3 GHz clock frequency. Only the BM3D was executed on a server with an Intel Core i7-7700 processor with 3.6 GHz clock frequency. Table IV summarizes the last-mentioned two aspects in form of a grading appraisal.

V. DISCUSSION AND CONCLUSION

In summary, none of the considered filters achieves a perfect result for every measurement. Nevertheless, the BM3D and the RGF are good candidates for the edge-preserving noise reduction in CSDs during the automated tuning of double quantum dots. The BM3D achieves high-quality metric scores and performs very well for datasets with heavy Gaussian-like noise. Generally, the RGF creates very sharp edges being helpful for the later detection of the transition lines in the CSDs. Moreover, it removes RTN, but for solid Gaussian noise, the results deteriorate. Furthermore, the BF can attain promising results for datasets with low noise and high-contrast edges, but these conditions do not appear often. In conclusion, we suggest introducing the BM3D into the automated tuning processing pipeline due to its striking quality, minimal effort for parameter determination, and acceptable runtime that can be even improved with an optimized implementation. However, if much RTN corrupts the data and dedicated algorithms have not reduced it before, the BF and the RGF are also good choices. When taking the scalability of a quantum computer into account, classical signal and image processing solutions have the advantage of enabling a hardware-closer implementation due to lower complexity. Their robustness is enhanced if one of the above methods

TABLE V Configurations for the Images for the First Gate Layout

filename	B1 [V]	P1 [V]	B2 [V]	P2 [V]	B3 [V]	P3 [V]	B4 [V]	P4 [V]	B5 [V]
GL1-01	-0.461153	*sweep*	-0.480151	*sweep*	-0.574985	0.000000	0.000000	0.000000	0.000000
GL1-02	-0.461153	*sweep*	-0.480151	*sweep*	-0.574985	0.000000	0.000000	0.000000	0.000000
GL1-03	-0.456321	*sweep*	-0.478447	*sweep*	-0.547163	-0.049999	-0.190000	-0.049999	-0.134991
GL1-04	-0.456321	*sweep*	-0.438366	*sweep*	-0.547163	-0.049999	-0.190000	-0.049999	-0.134991
GL1-05	-0.415223	*sweep*	-0.273035	*sweep*	-0.476997	-0.049999	-0.190000	-0.049999	-0.134991
GL1-06	-0.415223	*sweep*	-0.273035	*sweep*	-0.476997	-0.049999	-0.190000	-0.049999	-0.134991
GL1-07	-0.452455	*sweep*	-0.432848	*sweep*	-0.485288	-0.049999	-0.299992	-0.049999	-0.299992
GL1-08	-0.452455	*sweep*	-0.432848	*sweep*	-0.485288	-0.049999	-0.299992	-0.049999	-0.299992
GL1-09	-0.452455	*sweep*	-0.432848	*sweep*	-0.485288	-0.049999	-0.299992	-0.049999	-0.299992
GL1-11	-0.452455	*sweep*	-0.432848	*sweep*	-0.485288	-0.049999	-0.299992	-0.049999	-0.299992
GL1-12	-0.462094	*sweep*	-0.448564	*sweep*	-0.499987	-0.049999	-0.299992	-0.049999	-0.299992
GL1-13	-0.462094	*sweep*	-0.448564	*sweep*	-0.499987	-0.049999	-0.299992	-0.049999	-0.299992
GL1-14	-0.439434	*sweep*	-0.392284	*sweep*	-0.512652	-0.049999	-0.249994	-0.049999	-0.249994
GL1-15	-0.439434	*sweep*	-0.392284	*sweep*	-0.512652	-0.049999	-0.249994	-0.049999	-0.249994
GL1-16	-0.414206	*sweep*	-0.351136	*sweep*	-0.456194	-0.049999	-0.249994	-0.049999	-0.249994
GL1-17	-0.412299	*sweep*	-0.328527	*sweep*	-0.500445	-0.049999	-0.199995	-0.049999	-0.199995
GL1-18	-0.412782	*sweep*	-0.313013	*sweep*	-0.490527	-0.049999	-0.199995	-0.049999	-0.199995
GL1-19	-0.249994	-0.049999	-0.249994	-0.049999	-0.410010	*sweep*	-0.443300	*sweep*	-0.486941
GL1-20	-0.249994	-0.049999	-0.249994	-0.049999	-0.466189	*sweep*	-0.403194	*sweep*	-0.473538
GL1-21	-0.199995	-0.049999	-0.199995	-0.049999	-0.504463	*sweep*	-0.479922	*sweep*	-0.409934
GL1-22	-0.199995	-0.049999	-0.199995	-0.049999	-0.490730	*sweep*	-0.477963	*sweep*	-0.422192
GL1-23	-0.199995	-0.049999	-0.199995	-0.049999	-0.469088	*sweep*	-0.406628	*sweep*	-0.457923
GL1-24	-0.199995	-0.049999	-0.199995	-0.049999	-0.465629	*sweep*	-0.412273	*sweep*	-0.465324
GL1-25	-0.199995	-0.049999	-0.199995	-0.049999	-0.465629	*sweep*	-0.412273	*sweep*	-0.465324

filename	SL1 [V]	SP1 [V]	SR1 [V]	SL2 [V]	SP2 [V]	SR2 [V]	left or right SD	image range
GL1-01	-0.633783	-0.046845	-0.449454	-0.599374	-0.039267	-0.700008	left	[-0.1918, -0.1195]
GL1-02	-0.635004	-0.047176	-0.450853	-0.449989	-0.049999	-0.449989	left	[0.5753, 0.6195]
GL1-03	-0.631698	-0.037054	-0.445411	-0.596475	-0.043081	-0.603927	left	[-0.1976, -0.1214]
GL1-04	-0.631698	-0.037054	-0.445411	-0.596475	-0.043081	-0.603927	right	[-0.2666, -0.1420]
GL1-05	-0.635030	-0.045116	-0.441291	-0.613845	-0.043565	-0.628926	left	[-0.1904, -0.1500]
GL1-06	-0.635030	-0.045116	-0.441291	-0.613845	-0.043565	-0.628926	right	[-0.3369, -0.2523]
GL1-07	-0.634521	-0.093309	-0.443198	-0.592152	-0.038224	-0.603367	left	[-0.0991, -0.0653]
GL1-08	-0.634521	-0.093309	-0.443198	-0.592152	-0.038224	-0.603367	right	[-0.3154, -0.2966]
GL1-09	-0.634521	-0.090537	-0.443198	-0.592152	-0.038224	-0.603367	left	[-0.1384, -0.0837]
GL1-10	-0.634521	-0.090537	-0.443198	-0.592152	-0.038224	-0.603367	right	[-0.2488, -0.2326]
GL1-11	-0.636301	-0.084357	-0.449785	-0.594593	-0.046921	-0.597823	left	[-0.0944, -0.0527]
GL1-12	-0.636301	-0.084357	-0.449785	-0.594593	-0.046921	-0.597823	right	[-0.3100, -0.3016]
GL1-13	-0.626129	-0.100735	-0.443122	-0.601256	-0.082729	-0.610030	left	[-0.2046, -0.0531]
GL1-14	-0.626129	-0.100735	-0.443122	-0.601256	-0.082729	-0.610030	right	[-0.3463, -0.3356]
GL1-15	-0.559726	-0.021871	-0.535108	-0.618499	-0.045777	-0.599273	left	[-0.2873, -0.1276]
GL1-16	-0.545332	-0.046871	-0.524249	-0.616999	-0.046871	-0.599934	right	[-0.3200, -0.1504]
GL1-17	-0.545281	-0.046235	-0.524351	-0.616999	-0.046871	-0.599934	left	[-0.1271, -0.0913]
GL1-18	-0.555556	-0.027110	-0.544188	-0.600290	-0.049592	-0.603876	right	[-0.3078, -0.1722]
GL1-19	-0.555556	-0.027110	-0.544188	-0.600290	-0.049592	-0.603876	left	[-0.3027, -0.2089]
GL1-20	-0.556242	-0.051219	-0.539254	-0.602681	-0.042878	-0.590346	left	[-0.3168, -0.2800]
GL1-21	-0.556242	-0.051219	-0.539254	-0.602681	-0.042878	-0.590346	left	[-0.3170, -0.2790]
GL1-22	-0.556242	-0.047684	-0.539254	-0.599298	-0.040767	-0.593042	left	[-0.3126, -0.2076]
GL1-23	-0.556242	-0.047684	-0.539254	-0.599298	-0.042674	-0.593042	right	[-0.2475, -0.1694]
GL1-24	-0.556242	-0.047684	-0.539254	-0.599298	-0.042674	-0.593042	left	[-0.2758, -0.2175]

is applied before. However, machine learning methods, often used in automated tuning, also may benefit from their use. In general, they do not comprise such dedicated noise removal mechanisms directly, but we expect an improved performance when applying these methods to the training and experimental data before, especially if the training dataset is small. As the charge stability diagrams of other semiconductor qubit realizations are similar to those that we examined in this study, our results are also transferable to their autotuning process.

APPENDIX A METRIC RESULTS FOR ADDITIONAL RANDOM TELEGRAPH NOISE

See Fig. 17.

APPENDIX B GATE VOLTAGES AND MEASUREMENT RANGES FOR CSDS FROM GL1

The gate names for the first layout can be found in Fig. 1(a). The gates that are swept have voltages between −0.015 and 0.015 and are marked by *sweep* in Table V.

APPENDIX C GATE VOLTAGES AND MEASUREMENT RANGES FOR CSDS FROM GL2

The gate names of the second layout can be found in Fig. 1(b). The gates RFC and RFD are swept and have voltages between −0.0075 and 0.0075 V. The gates RFA and RFB do not have any voltage applied. Gate TAB has a voltage of −0.469978 V, Gate LP a voltage of −0.243076, and Gate RP a voltage of −0.243076 for all datasets. See Table VI.

TABLE VI Configurations for the Images for the Second Gate Layout

filename	SA	NAB	NBC	NCD	SD	PA	PB	TBC	PC
GL2-01	-0.700008	-0.809008	-1.099997	-0.950001	-0.779126	-0.733018	-0.299916	-1.050914	-0.300018
GL2-02	-0.700008	-0.769004	-1.099997	-0.829989	-0.779126	-0.783017	-0.299916	-1.050914	-0.300018
GL2-03	-0.700008	-0.795783	-1.019659	-0.905369	-0.770199	-0.814043	-0.331146	-0.886269	-0.286234
GL2-04	-0.717861	-0.792808	-0.995651	-0.866356	-0.770199	-0.796190	-0.431144	-0.886269	-0.311208
GL2-05	-0.717861	-0.792808	-0.995651	-0.866356	-0.770199	-0.796190	-0.431144	-0.886269	-0.311208
GL2-06	-0.717861	-0.771979	-0.995651	-0.899367	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-07	-0.717861	-0.771979	-0.995651	-0.899367	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-08	-0.717861	-0.771979	-0.995651	-0.899367	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-09	-0.717861	-0.771979	-0.995651	-0.899367	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-10	-0.717861	-0.771979	-0.995651	-0.890364	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-11	-0.717861	-0.771979	-0.995651	-0.890364	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-12	-0.717861	-0.771979	-0.995651	-0.890364	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-13	-0.717861	-0.771979	-0.995651	-0.890364	-0.770199	-0.796190	-0.431144	-0.886269	-0.438163
GL2-14	-0.722820	-0.771979	-0.995651	-0.890364	-0.790036	-0.787264	-0.431144	-0.886269	-0.438163
GL2-15	-0.722820	-0.771979	-0.995651	-0.890364	-0.790036	-0.787264	-0.431144	-0.886269	-0.438163

filename	TCD	PD	LT	LB	RT	RB	left or right SD	image range
GL2-01	-0.500013	-0.695099	-0.605580	-0.798530	-0.662343	-0.689377	right	[0.1939, 0.3284]
GL2-02	-0.500013	-0.745098	-0.622721	-0.757432	-0.687699	-0.646525	left	[0.3093, 0.3800]
GL2-03	-0.457364	-0.745098	-0.623891	-0.729254	-0.694285	-0.640447	right	[0.2027, 0.3377]
GL2-04	-0.457364	-0.745098	-0.616464	-0.735892	-0.692404	-0.637039	left	[0.2364, 0.3014]
GL2-05	-0.457364	-0.745098	-0.616464	-0.735892	-0.692404	-0.637039	right	[0.1958, 0.3378]
GL2-06	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	left	[0.2094, 0.2939]
GL2-07	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	right	[0.1867, 0.3362]
GL2-08	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	left	[0.2087, 0.2936]
GL2-09	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	right	[0.1859, 0.3351]
GL2-10	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	left	[0.2037, 0.2911]
GL2-11	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	right	[0.1849, 0.3344]
GL2-12	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	left	[0.2253, 0.2852]
GL2-13	-0.457364	-0.670710	-0.610920	-0.728542	-0.679510	-0.728313	right	[0.1804, 0.3342]
GL2-14	-0.457364	-0.670710	-0.605961	-0.737469	-0.639328	-0.733272	left	[0.2215, 0.2892]
GL2-15	-0.457364	-0.670710	-0.605961	-0.737469	-0.639328	-0.733272	right	[0.2706, 0.3320]

REFERENCES

[1] D. Loss and D. P. DiVincenzo, “Quantum computation with quantum dots,” *Phys. Rev. A*, vol. 57, no. 1, pp. 120–126, Jan. 1998, doi: 10.1103/PhysRevA.57.120.

[2] K. C. Nowack, F. H. L. Koppens, Yu V. Nazarov, and L. M. K. Vandersypen, “Coherent control of a single electron spin with electric fields,” *Science*, vol. 318, no. 5855, pp. 1430–1433, Nov. 2007, doi: 10.1126/science.1148092.

[3] S. Nadj-Perge, S. M. Frolov, E. P. A. M. Bakkers, and L. P. Kouwenhoven, “Spin-orbit qubit in a semiconductor nanowire,” *Nature*, vol. 468, no. 7327, pp. 1084–1087, Dec. 2010, doi: 10.1038/nature09682.

[4] J. R. Petta *et al.*, “Coherent manipulation of coupled electron spins in semiconductor quantum dots,” *Science*, vol. 309, no. 5744, pp. 2180–2184, Sep. 2005, doi: 10.1126/science.1116955.

[5] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen, “Spins in few-electron quantum dots,” *Rev. Mod. Phys.*, vol. 79, no. 4, pp. 1217–1265, Oct. 2007, doi: 10.1103/RevModPhys.79.1217.

[6] T. Fujita, T. A. Baart, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen, “Coherent shuttle of electron-spin states,” *NPJ Quantum Inf.*, vol. 3, no. 1, pp. 1–6, Jun. 2017, doi: 10.1038/s41534-017-0024-4.

[7] P. Cerfontaine, “High-fidelity single- and two-qubit gates for two-electron spin qubits,” Ph.D. dissertation, Dept. Fac. Math., Comput. Sci. Natural Sci., RWTH Aachen Univ., Aachen, Germany, Jul. 2019, doi: 10.18154/RWTH-2019-09348.

[8] C. Volk *et al.*, “Loading a quantum-dot based ‘qubyte’ register,” *NPJ Quantum Inf.*, vol. 5, no. 1, Dec. 2019, Art. no. 29, doi: 10.1038/s41534-019-0146-y.

[9] T. Botzem, “Coherence and high fidelity control of two-electron spin qubits in GaAs quantum dots,” Ph.D. dissertation, Dept. Fac. Math., Comput. Sci. Natural Sci., RWTH Aachen Univ, Aachen, Germany, 2017, doi: 10.18154/RWTH-2017-04410.

[10] W. S. Geisler, “Visual perception and the statistical properties of natural scenes,” *Annu. Rev. Psychol.*, vol. 59, no. 1, pp. 167–192, 2008, doi: 10.1146/annurev.psych.58.110405.085632.

[11] J. Darulova, M. Troyer, and M. C. Cassidy, “Evaluation of synthetic and experimental training data in supervised machine learning applied to charge state detection of quantum dots,” *Mach. Learning: Sci. Technol.*, vol. 2, no. 4, 2021, Art. no. 045023, doi: 10.1103/PhysRevApplied.13.054019.

[12] S. S. Kalantre *et al.*, “Machine learning techniques for state recognition and auto-tuning in quantum dots,” *NPJ Quantum Inf.*, vol. 5, no. 1, Dec. 2019, Art. no. 6, doi: 10.1038/s41534-018-0118-7.

[13] R. Durrer *et al.*, “Automated tuning of double quantum dots into specific charge states using neural networks,” *Phys. Rev. Appl.*, vol. 13, no. 5, May 2020, Art. no. 054019, doi: 10.1103/PhysRevApplied.13.054019.

[14] J. P. Zwolak *et al.*, “Autotuning of double-dot devices in situ with machine learning,” *Phys. Rev. Appl.*, vol. 13, no. 3, Mar. 2020, Art. no. 34075, doi: 10.1103/PhysRevApplied.13.034075.

[15] J. Darulová *et al.*, “Autonomous tuning and charge state detection of gate defined quantum dots,” *Phys. Rev. Appl.*, vol. 13, no. 5, May 2020, Art. no. 054005, doi: 10.1103/PhysRevApplied.13.054005.

[16] S. Czischek *et al.*, “Miniaturizing neural networks for charge state autotuning in quantum dots,” *Mach. Learn., Sci. Technol.*, vol. 3, 2021, Art. no. 015001, doi: 10.1088/2632-2153/AC34DB.

[17] M. Lapointe-Major *et al.*, “Algorithm for automated tuning of a quantum dot into the single-electron regime,” *Phys. Rev. B*, vol. 102, no. 8, Aug. 2020, Art. no. 85301, doi: 10.1103/PhysRevB.102.085301.

[18] L. A. F. Fernandes and M. M. Oliveira, “Real-time line detection through an improved hough transform voting scheme,” *Pattern Recognit.*, vol. 41, no. 1, pp. 299–314, 2008, doi: 10.1016/j.patcog.2007.04.003.

[19] C. Akinlar and C. Topal, “Edlines: A real-time line segment detector with a false detection control,” *Pattern Recognit. Lett.*, vol. 32, no. 13, pp. 1633–1642, 2011, doi: 10.1016/j.patrec.2011.06.001.

[20] H. Moon *et al.*, “Machine learning enables completely automatic tuning of a quantum device faster than human experts,” *Nature Commun.*, vol. 11, no. 1, Dec. 2020, Art. no. 4161, doi: 10.1038/s41467-020-17835-9.

[21] T. Botzem *et al.*, “Tuning methods for semiconductor spin qubits,” *Phys. Rev. Appl.*, vol. 10, no. 5, Nov. 2018, Art. no. 054026, doi: 10.1103/PhysRevApplied.10.054026.

[22] A. R. Mills, D. M. Zajac, M. J. Gullans, F. J. Schupp, T. M. Hazard, and J. R. Petta, "Shuttling a single charge across a one-dimensional array of silicon quantum dots," *Nature Commun.*, vol. 10, no. 1, Mar. 2019, Art. no. 1063, doi: 10.1038/s41467-019-08970-z.

[23] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, "Processing of RI-Angiocardiographic images," in *Digital Processing of Biomedical Images*, K. Preston and M. Onoe, Eds. Boston, MA, USA: Springer, 1976, pp. 187–202.

[24] M. Nagao and T. Matsuyama, "Edge preserving smoothing," *Comput. Graph. Image Process.*, vol. 9, no. 4, pp. 394–407, 1979, doi: 10.1016/0146-664X(79)90102-3.

[25] L. Davies and A. Rosenfeld, "Noise cleaning by iterated local averaging," *IEEE Trans. Syst., Man, Cybern.*, vol. 8, no. 9, pp. 705–710, Sep. 1978, doi: 10.1109/TSMC.1978.4310057.

[26] J. Lee, "Digital image enhancement and noise filtering by use of local statistics," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2 no. 2, pp. 165–168, Mar. 1980, doi: 10.1109/TPAMI.1980.4766994.

[27] Y. Lee and S. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 3, pp. 672–683, Jun. 1985, doi: 10.1109/TASSP.1985.1164591.

[28] S. Smith and M. Brady, "Susan—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, 1997, doi: 10.1023/A:1007963824710.

[29] S. Paris, S. Hasinoff, and J. Kautz, "Local Laplacian filters: Edge aware image processing with a Laplacian pyramid," *ACM Trans. Graph.*, vol. 30, no. 4, 2011, Art. no. 68, doi: 10.1145/2723694.

[30] V. Frost, J. Stiles, K. Shanmugan, and J. Holtzman, "A model for radar images and its application to adaptive digital filtering of multiplicative noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-4, no. 2, pp. 157–166, Mar. 1982, doi: 10.1109/TPAMI.1982.4767223.

[31] J.-S. Lee, "Digital image smoothing and the sigma filter," *Comput. Vis. Graph. Image Process.*, vol. 24, pp 255–269, 1983, doi: 10.1016/0734-189X(83)90047-6.

[32] D. Kuan, A. Sawchuk, T. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal-dependent noise," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 7, no. 2, pp. 165–77, Mar. 1985, doi: 10.1109/TPAMI.1985.4767641.

[33] A. Baraldi and F. Parmiggiani, "A refined gamma MAP SAR speckle filter with improved geometrical adaptivity," *IEEE Trans. Geosci. Remote Sens.*, vol. 33, no. 5, pp. 1245–1257, Sep. 1995, doi: 10.1109/36.469489.

[34] P. Saint-Marc, J. Chen, and G. Medioni, "Adaptive smoothing: A general tool for early vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 514–529, Jun. 1991, doi: 10.1109/34.87339.

[35] T. Qiu, A. Wang, N. Yu, and A. Song, "Llsure: Local linear sure-based edge-preserving image filtering," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 80–90, Jan. 2013, doi: 10.1109/TIP.2012.2214052.

[36] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990, doi: 10.1109/34.56205.

[37] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, Mar. 1998, doi: 10.1109/83.661192.

[38] J. Weickert, B. M. T. H. Romeny, and M. A. Viergever, "Efficient and reliable schemes for nonlinear diffusion filtering," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 398–410, Mar. 1998, doi: 10.1109/83.661190.

[39] V. Aurich and J. Weule, "Non-linear Gaussian filters performing edge preserving diffusion," in *Mustererkennung*, G. Sagerer, S. Posch, and F. Kummert, Eds., Berlin, Germany: Springer, 1995, pp. 538–545, doi: 10.1007/978-3-642-79980-8_63.

[40] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846, doi: 10.1109/ICCV.1998.710815.

[41] T. Pham and L. van Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proc. IEEE Int. Conf. Multimedia Expo*, Amsterdam, The Netherlands, 2005, pp. 454–457, doi: 10.1109/ICME.2005.1521458.

[42] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proc. 29th Annu. Conf. Comput. Graph. Interactive Techn.*, San Antonio, TX, USA, 2002, Art. no. 257, doi: 10.1145/566654.566574.

[43] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. Eur. Conf. Comput. Vis.*, A. Leonardis, H. Bischof, and A. Pinz, Eds., 2006, vol. 3954, pp. 568–580, doi: 10.1007/11744085_44.

[44] K. N. Chaudhury, D. Sage, and M. Unser, "Fast $O(1)$ bilateral filtering using trigonometric range kernels," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376–3382, Dec. 2011, doi: 10.1109/TIP.2011.2159234.

[45] M. Zhang and B. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324–2333, Dec. 2008, doi: 10.1109/TIP.2008.2006658.

[46] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *Proc. ACM SIGGRAPH Courses*, 2005, pp. 5–es, doi: 10.1145/1198555.1198565.

[47] R. Garnett, T. Huegerich, C. Chui, and W. He, "A universal noise removal algorithm with an impulse detector," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp 1747–54, Nov. 2005, doi: 10.1109/TIP.2005.857261.

[48] Y.-H. Liu, K. Gao, G.-G. Ni, and S.-L. Ge, "A universal denoising algorithm with trilateral filter and impulse detector," in *Proc. Int. Conf. Opt. Instrum. Technol., Optoelectron. Imag. Process Technol.*, T. Yoshizawa, P. Wei, and J. Zheng, Eds., 2009, vol. 7513, pp. 646–655, doi: 10.1117/12.838118.

[49] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013, doi: 10.1109/TPAMI.2012.213.

[50] K. He and J. Sun, "Fast guided filter," 2015, *arXiv:1505.00996*, doi: 10.48550/arXiv.1505.00996.

[51] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 120–129, Jan. 2015, doi: 10.1109/TIP.2014.2371234.

[52] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *Proc. Eur. Conf. Comput. Vis.*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., 2014, vol. 8691, pp. 815–830, doi: 10.1007/978-3-319-10578-9_53.

[53] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhoods," *IEEE Signal Process. Lett.*, vol. 12 no. 12, pp. 839–842, Dec. 2005, doi: 10.1109/LSP.2005.859509.

[54] S. Awate and R. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 364–76, Mar. 2006, doi: 10.1109/TPAMI.2006.64.

[55] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, 2005, vol. 2, pp. 60–65, doi: 10.1109/CVPR.2005.38.

[56] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006, doi: 10.1109/TIP.2006.877529.

[57] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006, doi: 10.1109/TIP.2006.881969.

[58] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D, Nonlinear Phenomena*, vol. 60, no. 1/4, pp. 259–268, Nov. 1992, doi: 10.1016/0167-2789(92)90242-F.

[59] S. Osher, W. Yin, D. Goldfarb, and J. Xu, "An iterative regularization method for total variation-based image restoration," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 460–89, Jan. 2005, doi: 10.1137/040605412.

[60] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Model. Simul.*, vol. 7, pp 1005–1028, Jan. 2008, doi: 10.1137/070698592.

[61] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, Aug. 2008, Art. no. 67, doi: 10.1145/1360612.1360666.

[62] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," in *Proc. SIGGRAPH Asia Conf.*, 2011, pp. 1–12, doi: 10.1145/2070781.2024208.

[63] D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. Do, "Fast global image smoothing based on weighted least squares," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5638–5653, Dec. 2014, doi: 10.1109/TIP.2014.2366600.

[64] D. Donoho and I. Johnstone, "Ideal spatial adaptation by wavelet Shrinkage," *Biometrika*, vol. 81, pp 425–455, 1994, doi: 10.2307/2337118.

[65] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, May 1995, doi: 10.1109/18.382009.

[66] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, Dec. 1995, doi: 10.2307/2291512.

[67] H. Chipman, E. Kolaczyk, and R. McCulloch, "Adaptive Bayesian wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 92, no. 440, pp. 1413–1421, 1997, doi: 10.2307/2965411.

[68] L. Sendur and I. W. Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Trans. Signal Process.*, vol. 50, no. 11, pp. 2744–2756, Nov. 2002, doi: 10.1109/TSP.2002.804091.

[69] G. Y. Chen, T. D. Bui, and A. Krzyżak, "Image denoising with neighbour dependency and customized wavelet and threshold," *Pattern Recognit.*, vol. 38, no. 1, pp. 115–124, 2005, doi: 10.1016/j.patcog.2004.05.009.

[70] N. Weyrich and G. T. Warhola, "Wavelet shrinkage and generalized cross validation for image denoising," *IEEE Trans. Image Process.*, vol. 7, no. 1, pp. 82–90, Jan. 1998, doi: 10.1109/83.650852.

[71] R. da Silva, R. Minetto, W. Schwartz, and H. Pedrini, "Adaptive edge-preserving image denoising using wavelet transforms," *Pattern Anal. Appl.*, vol. 16, pp 567–580, 2013, doi: 10.1007/s10044-012-0266-x.

[72] K. Dabov, A. Foi, V. R. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007, doi: 10.1109/TIP.2007.901238.

[73] F. Luisier, T. Blu, and M. Unser, "A new sure approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 593–606, Mar. 2007, doi: 10.1109/TIP.2007.891064.

[74] R. Fattal, "Edge-avoiding wavelets and their applications," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–10, 2009, doi: 10.1145/1531326.1531328.

[75] J. Ho and W. Hwang, "Wavelet Bayesian network image denoising," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1277–1290, Apr. 2013, doi: 10.1109/TIP.2012.2220150.

[76] B. Rieger, C. Luengo, and R. Ligteringen, "Dipimage," Accessed: Oct. 17, 2017. [Online]. Available: https://diplib.org/DIPimage.html

[77] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[78] E. Mühlhaus, "Die sprungerhaltende Glättung verrauschter, harmonischer schwingungen," Ph.D. dissertation, Dept. Fac. Math. Natural Sci., Heinrich Heine Univ., Düsseldorf, Germany, 1997.

[79] J. Immerkær, "Fast noise variance estimation," *Comput. Vis. Image Understanding*, vol. 64, no. 2, pp. 300–302, 1996, doi: 10.1006/cviu.1996.0060.

[80] P. Astola, 2013. [Online]. Available: http://freesourcecode.net/matlabprojects/62035/two-dimensional-trilateral-filter-in-matlab#.XsuJIzngqUk

[81] J. Manjon-Herrera and A. Buades, "Non-local means filter," Aug. 1, 2008. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter

[82] R. Pan and Rachel, "Guided filter," Oct. 6, 2011. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/33143-guided-filter

[83] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 664–672, Aug. 2004 doi: 10.1145/1015706.1015777.

[84] Q. Zhang, "Rolling guidance filter," 2014. [Online]. Available: http://www.cse.cuhk.edu.hk/leojia/projects/rollguidance/

[85] P. Getreuer, "Rudin-Osher-Fatemi total variation denoising using split Bregman," *Image Process.*, vol. 2, pp 74–95, May 2012, doi: 10.5201/ipol.2012.g-tvd.

[86] A. Mordvintsev, "ROF and TV-L1 denoising with primal-dual algorithm," 2017. [Online]. Available: https://www.webcitation.org/6rEjLnF1F

[87] A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*. ser., Radon Series on Computational and Applied Mathematics, M. Fornasier, Ed., Berlin, Germany: De Gruyter, 2010, pp. 263–340.

[88] M. Lourakis, "TV-L1 image denoising algorithm," Jun. 15, 2017. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/57604-tv-l1-image-denoising-algorithm

[89] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising with block-matching and 3D filtering," in *Image Processing: Algorithms and Systems, Neural Networks, and Machine Learning*, vol. 6064, N. M. Nasrabadi, S. A. Rizvi, E. R. Dougherty, J. T. Astola, and K. O. Egiazarian, Eds. Bellingham, WA, USA: SPIE, 2006, pp. 354–365, doi: 10.1117/12.643267.

[90] Y. Mäkinen *et al.*, "Image and video denoising by sparse 3D transform-domain collaborative filtering," 2019. [Online]. Available: https://webpages.tuni.fi/foi/GCF-BM3D/

[91] C. M. Stein, "Estimation of the mean of a multivariate normal distribution," *Ann. Statist.*, vol. 9, no. 6, pp. 1135–1151, 1981. [Online]. Available: https://www.jstor.org/stable/2240405

[92] R. Verma, "How to carry out sureshrink algorithm for wavelet based image denoising?," 2017. [Online]. Available: https://www.researchgate.net/post/How-to-carry-out-SureShrink-algorithm-for-wavelet-based-image-denoising

[93] davidji1/noise-qubit, Dec. 30, 2016. [Online]. Available: https://github.com/davidji1/noise-qubit

[94] M. Pedersen, "Full-reference image quality metrics: Classification and evaluation," *Found. Trends Comput. Graph. Vis.*, vol. 7, no. 1, pp. 1–80, 2011, doi: 10.1561/0600000037.

[95] B. Sankur, I. Avcibas, and K. Sayood, "Statistical evaluation of image quality measures," *J. Electron. Imag.*, vol. 11, no. 2, pp. 206–223, Apr. 2002, doi: 10.1117/1.1455011.

[96] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," *IEEE Signal Process. Mag.*, vol. 26, no. 1, pp. 98–117, Jan. 2009, doi: 10.1109/MSP.2008.930649.

[97] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

[98] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Comput. J.*, vol. 7, no. 4, pp. 308–313, Jan. 1965, doi: 10.1093/comjnl/7.4.308.

[99] J. D'Errico, Feb. 6, 2012. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/8277-fminsearchbnd-fminsearchcon

Sarah Fleitmann received the B.Sc. degree in scientific programming in 2020 from FH Aachen—University of Applied Sciences, Jülich, Germany, where she is currently working toward the M.Sc. degree in applied mathematics and informatics.

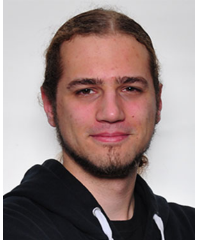
Since 2017, she has been working as a Software Developer with the Central Institute of Engineering, Electronics and Analytics—Electronic Systems, Forschungszentrum Jülich GmbH, Jülich, Germany. Her research interests include the automatic tuning of quantum dots for their operation as qubits.

Fabian Hader received the B.Sc. degree in scientific programming and the M.Sc. degree in energy economics and informatics from FH Aachen—University of Applied Sciences, Jülich, Germany, in 2019 and 2021, respectively. He is currently working toward the Ph.D. degree in engineering with University Duisburg-Essen, Duisburg/Essen, Germany.

From 2019 to 2021, he was a Software Engineer with the Central Institute of Engineering, Electronics and Analytics—Electronic Systems, Forschungszentrum Jülich GmbH, Jülich, Germany. His research interest focuses the automatic tuning of quantum dots.

Jan Vogelbruch received the Dipl.Ing. and Dr.-Ing. degrees in electrical engineering from RWTH Aachen University, Aachen, Germany, in 1994 and 2003, respectively.

Early 1995, he joined Parsytec Computer GmbH, Aachen, Germany, as a Technical Project Manager European cooperations. His focus has been in the area of high-performance computing and image processing solutions where he has been the technical leader for the company's part in several EC-funded projects. Since late 1998, he has been with the Central Institute of Engineering, Electronics and Analytics—Electronic Systems, Forschungszentrum Jülich GmbH, Jülich, Germany. His research interests include parallel computing, signal and 3-D image processing, fast reconstruction methods for high-resolution computer tomography, and automated defect detection. His current research focus is on the automatic tuning of semiconductor quantum dots.



Simon Humpohl received the B.Sc. and M.Sc. degrees in physics from RWTH Aachen University, Aachen, Germany in 2014 and 2017, respectively.

He is currently working toward the doctoral degree with the same university under the supervision of Prof. Hendrik Bluhm in the JARA Institute for Quantum Information. His main research interests are tuning and operation of electron spin qubits in semiconductor quantum dots.

Tobias Hangleiter received the B.Sc. and M.Sc. degree in physics from RWTH Aachen University, Aachen, Germany, in 2016 and 2019, respectively. He is currently working toward a doctoral degree in the field of quantum technology at the II. Institute of Physics.

His research interests include the automatic tuning of semiconductor quantum dots, quantum dynamics in the presence of correlated noise, and optical interfaces for semiconductor spin qubits.

Stefanie Meyer received the B.Sc. degree in scientific programming and the M.Sc. degree in technomathematics from FH Aachen—University of Applied Sciences, Jülich, Germany, in 2011 and 2013, respectively. In 2008, she joined the Institute of Energy and Climate Research–Fuel Cells, Forschungszentrum Jülich GmbH, Jülich, Germany.

Her research includes the development and parallelization of high temperature polymere electrolyte fuel cell models for HPC. Her research interests have a special focus on electronic systems for quantum computing. Since 2014, she has been the head of the software development team at the Central Institute of Engineering, Electronics and Analytics—Electronic Systems, Forschungszentrum Jülich GmbH, Jülich, Germany.

Stefan van Waasen received the diploma and doctoral degree in electrical engineering from Gerhard-Mercator-University, Duisburg, Germany, in 1994 and 1999, respectively. The topic of his doctoral thesis was optical receivers up to 60 Gb/s based on traveling wave amplifiers.

In 1998, he joined Siemens Semiconductors/Infineon Technologies AG, Düsseldorf, Germany. His responsibility was BiCMOS and CMOS RF system development for highly integrated cordless systems, such as DECT and Bluetooth. In 2001, he changed into the IC development of front-end systems for high data rate optical communication systems. From 2004 to 2006, he was with the Stockholm Design Center responsible for the short-range analog, mixed-signal, and RF development for SoC CMOS solutions. In the period 2006–2010, he was responsible for the wireless RF system engineering in the area of SoC CMOS products at the headquarter in Munich, Germany, and later in the Design Center Duisburg, Duisburg. Since 2010, he has been the Director of the Central Institute of Engineering, Electronics and Analytics—Electronic Systems, Forschungszentrum Jülich GmbH, Jülich, Germany. In 2014, he became a Professor for measurement and sensor systems at Communication Systems Chair of University Duisburg-Essen. Nowadays, his research is in the direction of complex measurement and detector systems with a special focus on electronic systems for quantum computing.