

Progressive Image Denoising

Claude Knaus, *Member, IEEE*, and Matthias Zwicker, *Member, IEEE*

Abstract—Image denoising continues to be an active research topic. Although state-of-the-art denoising methods are numerically impressive and approach theoretical limits, they suffer from visible artifacts. While they produce acceptable results for natural images, human eyes are less forgiving when viewing synthetic images. At the same time, current methods are becoming more complex, making analysis, and implementation difficult. We propose image denoising as a simple physical process, which progressively reduces noise by deterministic annealing. The results of our implementation are numerically and visually excellent. We further demonstrate that our method is particularly suited for synthetic images. Finally, we offer a new perspective on image denoising using robust estimators.

Index Terms—Image denoising, robust estimation, deterministic annealing, bilateral filtering, short-time Fourier transform.

I. INTRODUCTION

ONE of the most important problems of image processing is denoising, the reconstruction of the original image from a noisy image. Noisy images may be produced by noise contamination through an analog process during acquisition or transport over analog media. The common simplifying assumption is that the image has been contaminated with *additive white Gaussian noise* (AWGN). This assumption includes that the noise is stationary and uncorrelated among pixels. Another common assumption is that the variance of the noise is known.

Progress in image denoising has stagnated in recent years. The medal of state-of-the-art is held by *block-matching with 3D filtering* (BM3D) [1], aging over seven years. Levin and Nadler found that for natural images, BM3D is close to the theoretical limit of denoising [2], but artificial and highly correlated images still have potential for improvement [2]–[4]. However, only a handful of methods numerically improve over BM3D, with modest increase in visual quality. State-of-the-art image denoising methods still produce visible artifacts, especially on sharp edges and in smooth regions of the original image. Such features are common for natural images, like clear sky and human skin, not to speak of synthetic images, where edges and gradients are abundant. Another nuisance is that current methods are complex and thus prohibit thorough analysis.

Recently, Knaus and Zwicker demonstrated with *dual-domain image denoising* (DDID) that simple algorithms can

Manuscript received August 20, 2013; revised January 22, 2014 and May 13, 2014; accepted May 21, 2014. Date of publication May 23, 2014; date of current version June 16, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hitoshi Kiya.

The authors are with the University of Bern, Bern 3012, Switzerland (e-mail: knaus@iam.unibe.ch; zwicker@iam.unibe.ch).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2014.2326771

achieve high-quality results [5]. We extend their work and propose *progressive image denoising* (PID), a method inspired by deterministic annealing and based on robust noise estimation. Deterministic annealing (DA) is a heuristic method that is efficient at solving complex optimization problems where many local extrema exist. We combine deterministic annealing with redescending M-estimators, similar to previous works of Li [6] and Frühwirth and Waltenberger [7]. Our method produces high-quality results, void of artifacts typical to patch-based methods. It performs not only well for natural images, but also for synthetic images where artifacts are more apparent. It is also of practical interest that our algorithm is unusually short, fitting into a column of this paper. Last not least, our formulation using robust estimators and iterative filtering akin to deterministic annealing offers opportunities to explore alternative implementations of the image denoising process.

The rest of the paper is organized as follows. We first review state-of-the-art methods for image denoising and relate them to our contribution in Section II. In Section III, we explain our image denoising process using robust noise estimation and deterministic annealing. We elaborate the implementation details in Section IV and present high-quality results compared to state-of-the-art methods in Section V. Finally, in Section VI, we discuss the implications of understanding image denoising as robust noise estimation and conclude the paper in Section VII.

II. RELATED WORKS

Most state-of-the-art image denoising methods are based on denoising patches. The common approach is to cluster similar patches within a neighborhood window and to denoise them simultaneously. The denoised patches are then accumulated in a buffer, followed by normalization of every pixel by the number of overlapping patches.

The baseline for high-quality image denoising using patches remains *block matching with 3D filtering* (BM3D) [1]. BM3D stacks the collected patches on top of each other and denoises them by 3D-wavelet shrinkage. An improved version is BM3D-SAPCA [8], which first modifies the patches using polygonal shape masks and additionally performs a PCA to find a sparse representation of the patches. Even today, BM3D-SAPCA remains unchallenged in its numerical performance. More recently, *non-local Bayes* (NLB) [9] was proposed, which directly solves for the most likely patches by matrix inversion. For color images, NLB performs better than BM3D.

Other patch-based methods learn dictionaries for sparse representation of the patches. Currently, the best method using a single dictionary is *learned simultaneous sparse coding*

(LSSC) [10]. A more recent and even better method is *spatially adaptive iterative singular-value thresholding* (SAIST) [11], which recognizes a symmetry in local and non-local representation and identifies two dictionaries with the unitary matrices of the SVD. While LSSC is numerically better than BM3D, SAIST is also visually more pleasant as it produces less artifacts than most other methods.

Few state-of-the-art methods exist which deviate from the previously mentioned patch-framework. It has been shown that images can be denoised using simple artificial neural networks, in particular using *multi-layer perceptrons* (MLP) [12]. MLP learns a function that directly maps a noisy input patch to a denoised output patch. Unfortunately, adapting to new noise levels requires offline training, taking weeks of GPU-time.

Recently, another patch-less method was introduced called *dual-domain image denoising* (DDID) [5]. DDID produces results comparable to BM3D although the algorithm is exceptionally simple. It is an iterated and guided method that uses a combination of the bilateral filter [13] in the spatial domain and short-time Fourier transform (STFT) based wavelet shrinkage in the frequency domain.

Our method is derived from DDID, but differentiates itself by allowing iteration using arbitrary fine time steps and by not requiring a distinction between noisy and guide images. Furthermore, our approach offers a new perspective on denoising by considering it as robust noise estimation. Durand and Dorsey have analyzed the bilateral filter from the perspective of robust statistics [14], i.e., bias introducing signals are rejected as outliers. We extend this view to include the frequency domain by regarding wavelet shrinkage as robust noise estimation.

Finally, our method is inspired by deterministic annealing (DA), which is related to simulated annealing (SA). DA and SA are heuristic methods suited for complex optimization problems with a priori unknown global energy. SA has its origin in statistical mechanics, and was independently discovered by Kirkpatrick et al. [15] and Černý [16]. It is a powerful heuristic to solve optimization problems in a large state space with many local minima. It minimizes an energy by a random walk in the state space, similar to Metropolis sampling. Over time, the state transition space is shrunk, controlled by a falling temperature. Stochastic state transitions prevent the algorithm from getting stuck in local minima by allowing the energy to increase with some probability. An annealing schedule prescribes the slow descent of the temperature. Eventually, the state is frozen at the optimum. DA accelerates this cooling process by parametrizing the energy with the temperature: instead of stochastic state changes, the momentary energy is deterministically reduced at every step. In our work, we use DA but instead of minimizing momentary energy, we derive a gradient for descent.

Li et al. combined DA with robust statistics to introduce the *annealing M-estimator* [6]. The scale parameter of redescending M-estimators was replaced by a decreasing temperature parameter, making scale selection unnecessary. Frühwirth and Waltenberger [7] generalized the hybrid method further, allowing the shape of the estimator to change as a function of the temperature. SA has been used for image denoising

before, but with limited success [17]–[20]. Our approach, however, is based on DA and we achieve numerically and visually excellent results. We also explore the use of alternative redescending M-estimators.

III. PROGRESSIVE IMAGE DENOISING

Our image denoising method is based on deterministic annealing and robust noise estimation, and is implemented using a simple iterative filtering scheme. We describe the method in three parts. In Section III-A, we interpret image denoising as a gradient descent and identify the energy gradient with a noise differential, up to scale. Then, in Section III-B, we estimate the noise differential using redescending M-estimators in the spatial and frequency domains. Finally, we define the annealing schedule in Section III-C.

A. Progressive Noise Removal

Image denoising is an “ill-posed” problem. A signal x has been contaminated with additive white Gaussian noise n and variance σ^2 . The task is to decompose the noise contaminated signal y into its original signal x and noise instance n like

$$y = x + n. \quad (1)$$

In practice, we can only *estimate* a decomposition $y = \tilde{x} + \tilde{n}$. Many estimation problems are formulated as energy minimization problems. Thus, we attempt to formulate denoising as a gradient descent with

$$x_{i+1} = x_i - \lambda \nabla E(x_i), \quad (2)$$

starting at $x_0 = y$. The scale factor λ controls the step size in the direction of the gradient descent. An attempt to define the energy term as $E(x_i) = (x_i - x)^2$ fails, since x is unknown. However, we empirically discovered that the gradient of this energy, $\nabla E(x_i)$, can be estimated as a noise estimate n_i for iteration i . Substituting $\nabla E(x_i) \rightarrow n_i$, we get

$$x_{i+1} = x_i - \lambda n_i, \quad (3)$$

allowing us to reinterpret the gradient descent as a progressive removal of noise differentials λn_i , which integrate over time i to the estimated total noise instance as $\tilde{n} = \lambda \sum_{i=0}^{\infty} n_i$.

B. Robust Noise Estimation

To compute the noise estimate n_i for iteration i , we need to distinguish signal from noise. Conceptually, we decompose the noisy signal into three classes: large and medium amplitude signals and small amplitude noise. Large amplitude signals can be recognized in the spatial domain. For example, step signals have large gradients which are easily detected. When the amplitude of the signal is smaller and therefore more similar to the noise, signal and noise cannot be reliably distinguished in the spatial domain. However, we can use the property that signal is auto-correlated and noise is uncorrelated. Auto-correlated signals, i.e. waves, are best detected as large amplitudes in the frequency domain. By using robust estimators to reject large amplitude gradients in the spatial domain and medium amplitude waves in the frequency domain,

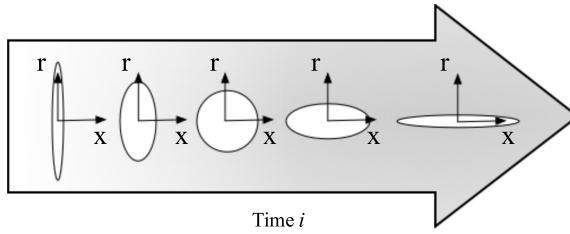


Fig. 1. Evolution of the bilateral kernel $k_r k_s$. Deterministic annealing changes the scale parameters by shrinking T_i and increasing S_i . The axis ‘r’ and ‘x’ represent the range and spatial dimensions.

we estimate the small amplitude noise without bias introduced by the signal. This entire process is illustrated in Fig. 3.

We derive now the robust noise estimator for time step i starting in the spatial domain, where we remove large amplitude signals. We consider pixel p using pixels q in a neighborhood window \mathcal{N}_p with window radius r . We first subtract the center pixel value $x_{i,p}$ from all the neighboring pixels $x_{i,q}$ yielding a “gradient” $d_{i,p,q}$ as

$$d_{i,p,q} = x_{i,q} - x_{i,p}. \quad (4)$$

Using this gradient, we define a smoothly decaying range kernel k_r to mask out large signals which have already been denoised. If those signals were kept, they would bias the noise estimate in the frequency domain. We also mask the signal by a smoothly decaying spatial kernel k_s , limiting bias from spatially distant pixels. Together, the two kernels k_r and k_s combine to an unnormalized bilateral kernel. Fig. 1 depicts the bandwidth of this bilateral kernel by ellipses. Finally, we perform a discrete Fourier transform to obtain the masked signal in the frequency domain \mathcal{F}_p , yielding the Fourier coefficients $D_{i,p,f}$ for frequency f as

$$D_{i,p,f} = \sum_{q \in \mathcal{N}_p} d_{i,p,q} k_r \left(\frac{|d_{i,p,q}|^2}{T_i} \right) \times k_s \left(\frac{|q-p|^2}{S_i} \right) e^{-j \frac{2\pi}{2r+1} f \cdot (q-p)}. \quad (5)$$

We used here the imaginary number $j = \sqrt{-1}$ to avoid confusion with the time i . The parameters T_i and S_i are scale parameters which we will describe in Section III-C.

In the frequency domain, we perform a similar procedure: we remove medium amplitude signals to obtain the remaining small amplitude noise. Thus, we use another range kernel K to mask out large Fourier coefficients $D_{i,p,f}$ representing fine structures and textures. Finally, we estimate the noise by taking the center pixel after inverse Fourier transforming the signal. To obtain this value, we apply the Fourier slicing theorem and average over all the Fourier coefficients and get

$$n_{i,p} = \frac{1}{(2r+1)^2} \sum_{f \in \mathcal{F}_p} D_{i,p,f} K \left(\frac{|D_{i,p,f}|^2}{V_i} \right). \quad (6)$$

Similar to the spatial domain, V_i is another scale parameter, which is in this case the variance of the Fourier coefficients

$D_{i,p,f}$, defined using the noise variance σ^2 as

$$V_i = \sigma^2 \sum_{q \in \mathcal{N}_p} k_r \left(\frac{|d_{i,p,q}|^2}{T_i} \right)^2 k_s \left(\frac{|q-p|^2}{S_i} \right)^2. \quad (7)$$

C. Shape Shifting Estimator

The missing ingredient to our method is the dynamic parameterization of the robust noise estimator. Analyzing DDID, we observed that the range parameter γ_r of the bilateral kernel exponentially decays over time, similar to deterministic annealing. Time-varying robust estimators have been used by Li et al. [6] to replace scale selection by DA. Frühwirth and Waltenberger [7] allowed other kernel shape parameters to change as well. While we shrink the scale of the range kernel k_r like in traditional DA, we simultaneously enlarge the spatial kernel k_s over time (Fig. 1). We define the scale parameters T_i and S_i as functions of time i :

$$T_i = \sigma^2 \gamma_r \alpha^{-i} \quad (8)$$

$$S_i = \sigma_s^2 \gamma_s \alpha^{i/2}. \quad (9)$$

The first scale parameter T_i of the range kernel k_r is our temperature which is reduced over time. We found that an exponential decay of the temperature works best, where α^{-1} is the rate of this decay. γ_r is a large initial scale factor. The second scale parameter S_i , however, we let grow. When the temperature T_i is high, the range kernel k_r covers the entire dynamic range and the spatial kernel k_s should be small to reduce bias from neighbouring pixels in the noise estimation. On the other hand, when the temperature is low, the range kernel becomes narrow and we require larger spatial support to discern autocorrelated signal with small amplitudes from noise. When the temperature has totally cooled down, the range kernel is a Dirac delta, and the spatial kernel is the constant 1, covering the entire spatial domain. The parameter σ^2 is the noise variance of the noisy input y and σ_s defines a reference standard deviation for the spatial kernel. Similar to the parameter γ_r of the range kernel, γ_s is a small initial scale factor for the scale σ_s of the spatial kernel.

IV. IMPLEMENTATION

Many denoising methods split their algorithm into two steps. First, an oracle is computed, and then the oracle is used to denoise the noisy image. The solution of deterministic annealing is only near-optimal as we may still end up in a local minimum. We therefore follow the same pattern like other methods and perform an additional denoising step. We discovered that the resulting image from the annealing stage serves as an excellent oracle to denoise the noisy signal with a single DDID step.

While we explored several redescending M-estimators for the range kernel, we settled to use Gaussians for all the kernels of the robust noise estimator, i.e., $k_r(d^2) = k_s(d^2) = K(d^2) = e^{-d^2}$. In Section VI, we analyze alternative robust estimators for the spatial range kernel k_r .

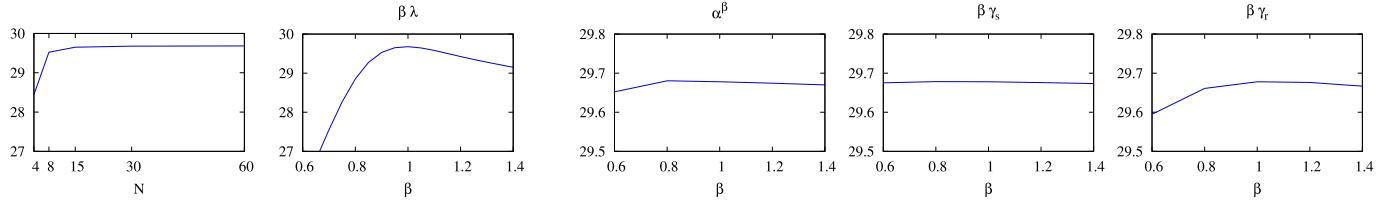


Fig. 2. Plots of PSNR (dB) for varying iteration number N and perturbed parameters $\beta\lambda$, α^β , $\beta\gamma_s$, and $\beta\gamma_r$. The plots were produced for *Cameraman*; plots for other images are similar. PID is robust with respect to changes of most parameters. The most influential parameters are N and λ , which control the noise estimation directly. The values chosen for the less influential parameters α , γ_s , γ_r are close to the optimal value. Changing them has little impact on the PSNR, in the range of 0.1 dB.

Algorithm 1 MATLAB Code for PID

```

function x = PID(y, sigma2)

N      = 30;
r      = 15;
sigma_s = 7;
gamma_r = 988.5;
gamma_s = 2/9;
alpha   = 1.533;
lambda  = log(alpha) * 0.567;
[dx dy] = meshgrid(-r:r);
r2     = dx.^2 + dy.^2;

x = y;
for i=1:N, xp = padarray(x, [r r], 'symmetric');
    parfor p=1:numel(y), [Y X] = ind2sub(size(y), p);

        % Spatial Domain
        d = xp(Y+2*r , X:X+2*r) - x(p);           % Eq. 4
        T = sigma2 * gamma_r * alpha^(i-1);          % Eq. 8
        S = sigma_s^2 * gamma_s * alpha^(i/2);         % Eq. 9
        k = exp(-d.^2 / T) .* exp(-r2 / S);          % Eq. 5

        % Fourier Domain
        D = fft2(ifftshift(d .* k));                 % Eq. 5
        V = sigma2 * sum(k(:).^2);                    % Eq. 7
        K = exp(-abs(D).^2 / V);                     % Eq. 6
        n = sum(sum(D .* K)) / numel(K);             % Eq. 6

        x(p) = x(p) - lambda * real(n);              % Eq. 3
    end
end

x = DDID_(x, y, sigma2, 31, 16, 0.6, 2.16);

```

Unlike most other image denoising methods, our algorithm is simple enough that we can provide a MATLAB implementation for grayscale images in this paper (Algorithm 1). For completeness, we also give the code of the DDID step (Algorithm 2). The formulation of the original DDID uses a wavelet shrinkage kernel in the frequency domain. To follow our paradigm, we replace wavelet shrinkage by robust noise estimation using $1 - e^{-\frac{D^2}{2.16V}} \approx e^{-\frac{V}{D^2}}$.

The parameters in the algorithm were empirically found and are the same for all noise levels. We use $N = 30$ iterations with a temperature decay rate of $\alpha^{-1} = 1.533^{-1}$ and gradient step size $\lambda = 0.567 \log \alpha$. These parameters change together. For example to double the number of steps, we would perform $N \rightarrow 2N$, $\alpha \rightarrow \sqrt{\alpha}$, and $\lambda \rightarrow \lambda/2$. The initial scale factor for the range scale is $\gamma_r = 988.5$, and for the spatial scale $\gamma_s = 2/9$. The window radius is $r = 15$, and we use a reference spatial sigma of $\sigma_s = 7$. For the final DDID step, we use a larger kernel size with window radius $r = 31$ and spatial sigma $\sigma_s = 16$. The range and frequency domain parameters are $\gamma_r = 0.6$ and $\gamma_s = 2.16$.

Algorithm 2 MATLAB Code of Original DDID Step. For Description, See Referenced Work [5]

```

function xt = DDID_(x, y, sigma2, r, sigma_s, gamma_r, gamma_f)

[dx dy] = meshgrid(-r:r);
h = exp(- (dx.^2 + dy.^2) / (2 * sigma_s^2));
xp = padarray(x, [r r], 'symmetric');
yp = padarray(y, [r r], 'symmetric');
xt = zeros(size(x));

parfor p = 1:numel(x), [i j] = ind2sub(size(x), p);

    % Spatial Domain: Bilateral Filter
    g = xp(i:i+2*r, j:j+2*r);
    y = yp(i:i+2*r, j:j+2*r);
    d = g - g(1:r, 1:r);
    k = exp(-d.^2 ./ (gamma_r * sigma2)) .* h;
    gt = sum(sum(g .* k)) / sum(k(:));
    st = sum(sum(y .* k)) / sum(k(:));

    % Fourier Domain: Wavelet Shrinkage
    V = sigma2 * sum(k(:).^2);
    G = fft2(ifftshift(g - gt) .* k);
    S = fft2(ifftshift((y - st) .* k));
    K = 1 - exp(-abs(G).^2 ./ (gamma_f * V));
    St = sum(sum(S .* K)) / numel(K);
    xt(p) = st + real(St);
end
end

```

In Fig. 2, we analyze the robustness of PID with respect to parameter change. Except for iteration number N , we modify the parameters by a perturbation value $0.6 \leq \beta \leq 1.4$. We plot the PSNR values as functions of iteration number N or perturbation value β . For this analysis, we used the *Cameraman* image and fixed the noise sigma to $\sigma = 25$. The plots for other images and noise levels are similar. The number of iterations N needs to be large enough to get good denoising results. The gradient descent step factor λ has the biggest influence on the PSNR. This is not surprising, since it has strong correlation with the noise estimation. Small values mean that the noise will be underestimated and consequently the image will contain residual noise. Conversely, large values mean that the noise is overestimated and the image will lose details. The remaining parameters are robust against change and are optimal with tolerance in the range of 0.1 dB.

Color images are handled similarly to DDID. We perform a 3-point discrete cosine transform (DCT) on the color channels. Since DCT is a unitary transformation, the noise variances remain constant and uncorrelated. When defining the range kernel, the normalized Euclidian distances are averaged over all channels. In the frequency domain, the channels are treated independently.

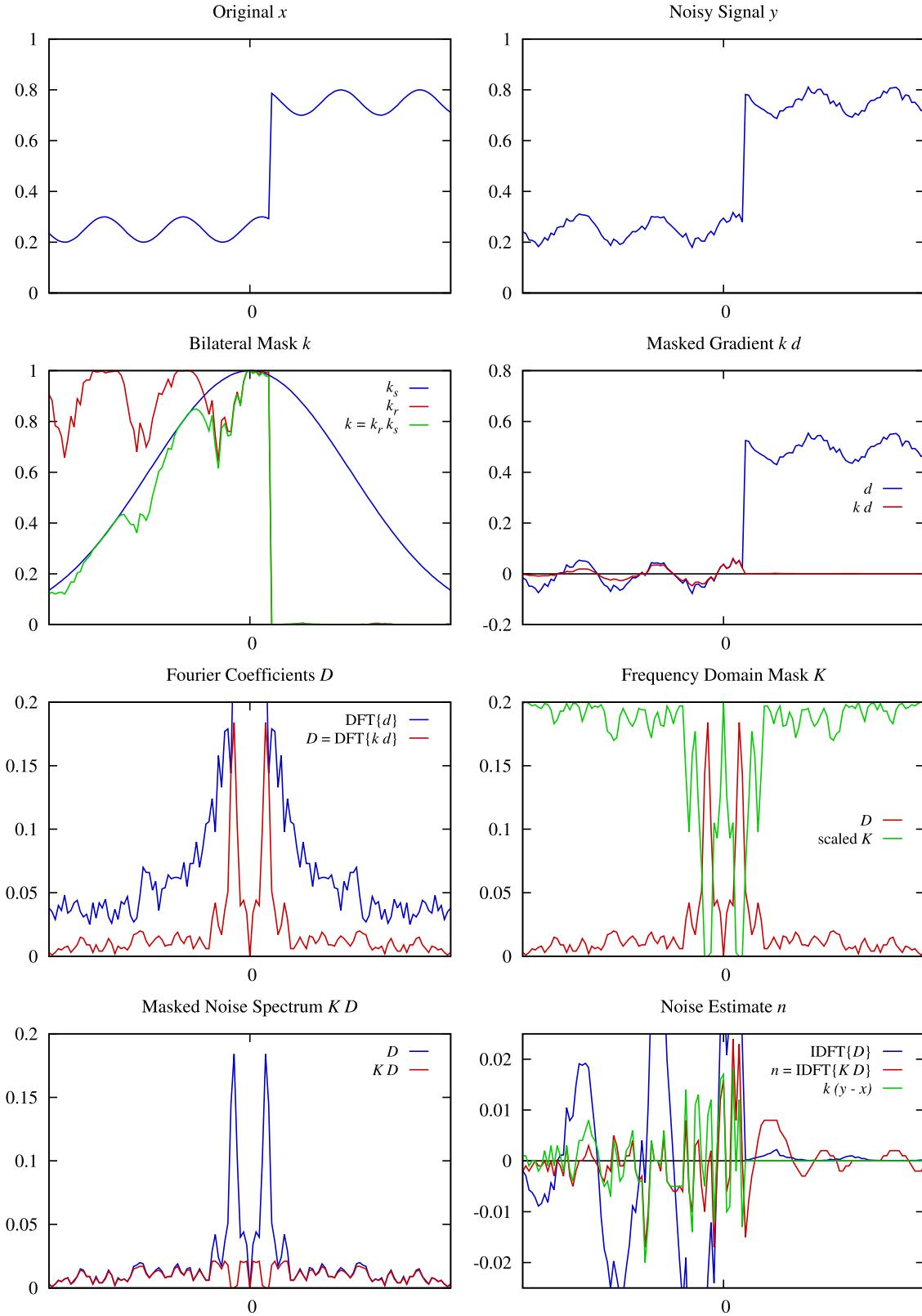


Fig. 3. Robust noise estimation from top left to bottom right. The original signal x has been contaminated with AWGN yielding the noisy signal y . Our goal is to estimate the noise at the origin. The green bilateral mask k is the product of the blue spatial Gaussian k_s and the red range Gaussian k_r . Multiplication with the blue gradient d gives the red masked gradient kd . If we compute the Fourier transform of the unmasked gradient d , we would get the blue unmasked Fourier coefficients $DFT\{d\}$, which are heavily biased by the large amplitude step. However, the Fourier transform of the bilaterally masked signal gives the red masked Fourier coefficients $D = DFT\{kd\}$. The detail waves are clearly visible and the green frequency domain mask K is calculated (up to scale). Multiplying the Fourier coefficients D with the mask K gives the red masked noise spectrum $K D$, which visibly contains only white Gaussian noise. Finally, the noise is estimated in the spatial domain by taking the inverse Fourier transform. The blue unmasked noise estimate $IDFT\{D\}$ is strongly biased by the detail waves, while the red masked noise estimate $IDFT\{KD\}$ correlates well around the origin with the green bilaterally masked ground truth noise $k(y - x)$.

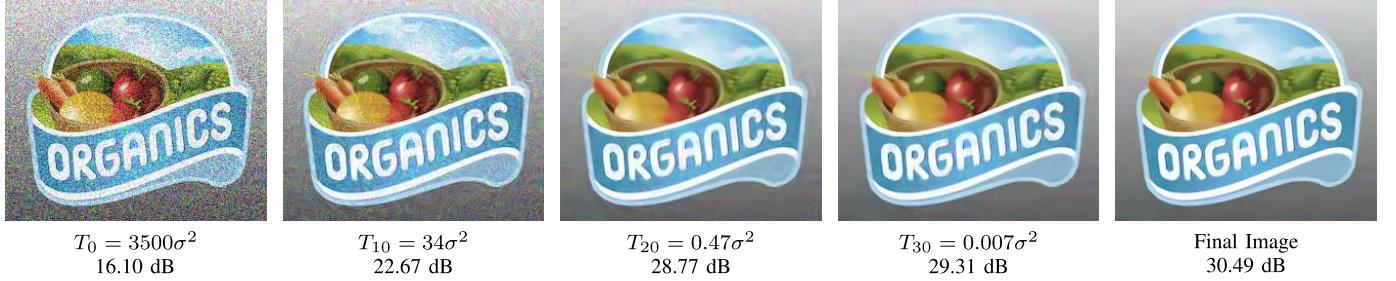


Fig. 4. Progressive denoising of a synthetic image by reducing the temperature T , given in units of the original noise variance σ^2 . The noise sigma is $\sigma = 40$. The second number denotes the increasing peak-signal-to-noise ratio (PSNR). All steps except for the last are calculated using explicit Euler integration; every image only depends on the previous one. The image at $T_{30} = 0.007\sigma^2$ is used as an oracle to produce the final image with a single DDID step.

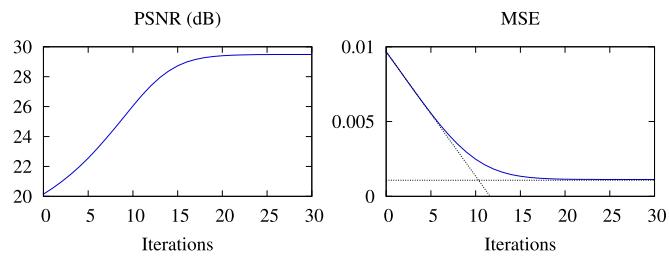


Fig. 5. Quality improvement over iterations for *Cameraman*. The PSNR increases nearly linearly for most of the time. The MSE follows a hyperbolic curve with the bias as one of its asymptotes.

V. RESULTS

We present the results of our algorithm. Fig. 4 displays the denoising process as an evolution starting with the noisy image. We used 30 iterations, where the intermediate images are snapshots taken after 10, 20, and 30 iterations. Usually, a denoising output of an iteration step cannot be used as input for another step, as the output pixels are correlated and estimating the variance would require expensive covariance tracking. In our case, however, correlated noise in the spatial domain is decorrelated in the frequency domain and therefore no covariance tracking is needed. This allows the iteration to go forward without getting stuck in a local minimum.

We analyze the denoising process for the familiar *Cameraman* (Fig. 7). Fig. 5 shows the improvement of quality over time. The PSNR increases fast in the beginning, and slows down as the noise becomes smaller. The corresponding MSE exhibits a strong linear decrease in the beginning and asymptotically approaching the squared bias as the variance vanishes. The plots look similar for any image we denoised.

We gain a better understanding about the annealing process by observing the evolution of pixels. In Fig. 6, we consider two pixels, x_q on the tall building and x_p in the sky, marked blue and red in Fig. 7. The difference is relatively small, in the range of the noise sigma, since $\sigma = 25/255 \approx 0.1$. In the first plot, we see the noise differentials of $n_{i,q}$ and $n_{i,p}$ as a function of time. Since the noise is uncorrelated, the noise differentials also take “independent” paths. They first rise steeply and then fall off, eventually vanishing. The second plot shows the integrals of the noise differentials, which are the range trajectories $x_{i,q}$ and $x_{i,p}$ of the pixels, starting with the noisy values y_q and y_p . We observe a smooth fall off towards the

TABLE I

PSNR (dB) COMPARISON OF ROBUST ESTIMATION KERNELS K_R FOR GRayscale IMAGES WITH NOISE SIGMA $\sigma = 25$. WELSCH IS THE MOST CONSERVATIVE DENOISER. EPANECHNIKOV IS THE MOST AGGRESSIVE DENOISER FOR IMAGES WITH HOMOGENEOUS REGIONS. TUKEY IS A COMPROMISE BETWEEN WELSCH AND EPANECHNIKOV

GRAYSCALE	BM3D	DDID	PID ^W	PID ^T	PID ^E
Barbara	30.72	30.80	30.56	30.70	30.79
Boats	29.91	29.79	29.80	29.81	29.78
Cameraman	29.45	29.46	29.68	29.64	29.55
Couple	29.72	29.56	29.64	29.66	29.62
Finger Print	27.70	27.32	27.17	27.25	27.31
Hill	29.85	29.71	29.77	29.75	29.68
House	32.86	32.66	32.84	32.90	32.95
Lena	32.08	32.14	32.12	32.17	32.18
Man	29.62	29.62	29.68	29.66	29.58
Montage	32.37	32.61	32.76	32.86	32.93
Pepper	30.16	30.26	30.34	30.39	30.40

^WWelsch ^TTukey ^EEpanechnikov

vicinity of their ground truth values, suggested by dashed lines. The third plot shows the relation between the two pixels as the squared difference/gradient $(x_{i,q} - x_{i,p})^2$ between them. For the first 15 iterations, this squared difference is smaller than the temperature T , so the squared gradient is considered as noise from the perspective of the spatial range kernel k_r . This means that the two pixels affect each other's noise estimates. After the two curves cross each other, the temperature is below the squared gradient and the interaction between the two pixels are “frozen”. This can be better seen in the last plot: the normalized Euclidean distance, i.e., the quotient

$$\frac{(x_{i,q} - x_{i,p})^2}{T_i}$$

becomes larger than 1 after iteration 15, which in turn lets the weight of the range kernel

$$e^{-\frac{(x_{i,q} - x_{i,p})^2}{T_i}}$$

vanish quickly. After iteration 20, the squared gradient between the pixels are considered outliers for both pixels.

A. Natural and Synthetic Images

PID produces aesthetically pleasing results. The first two examples in Fig. 11 demonstrate that our method works

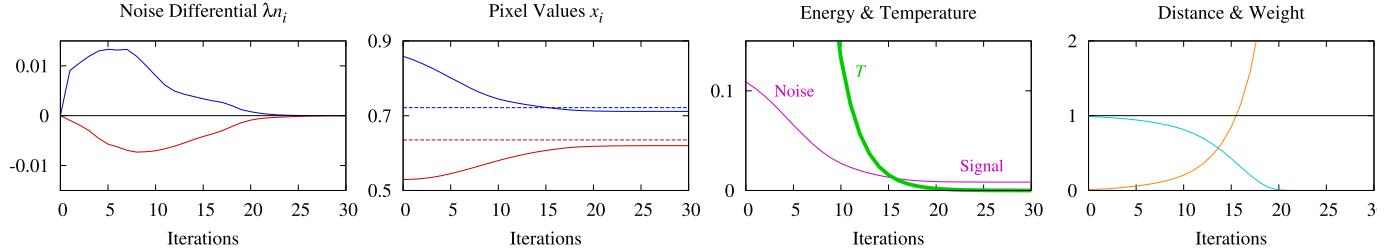


Fig. 6. Study of the evolution of two pixels across an edge on *Cameraman* (Fig. 7). Pixel q (blue) is on the tall building, pixel p (red) is in the sky, to the right of the building. The left most plot shows the evolution of the noise differentials $\lambda n_{i,q}$ and $\lambda n_{i,p}$ of the two pixels. They take individual paths and eventually converge to 0 when no change happens to either pixel. The second plot displays the pixel values $x_{i,q}$ and $x_{i,p}$, which are the time integrations of the noise differentials. The dashed lines are the ground truth values. The third plot displays the energy (purple), i.e., the squared difference $(x_{i,q} - x_{i,p})^2$ between the two pixels and the faster decreasing temperature T_i (green). The curves cross near iteration 15 at which point the interaction between the two pixels slows down. Up to this point the pixel difference was considered noise, after this point the pixel difference is considered signal. Note that the energy follows a similar curve as the MSE in Fig. 5. The last plot shows the quotient of the previous two curves, i.e., normalized Euclidean distance $\frac{(x_{i,q} - x_{i,p})^2}{T_i}$ (orange), and the Gaussian weight $e^{-\frac{(x_{i,q} - x_{i,p})^2}{T_i}}$ (cyan). After the 20th iteration, there is no more diffusion across the edge.

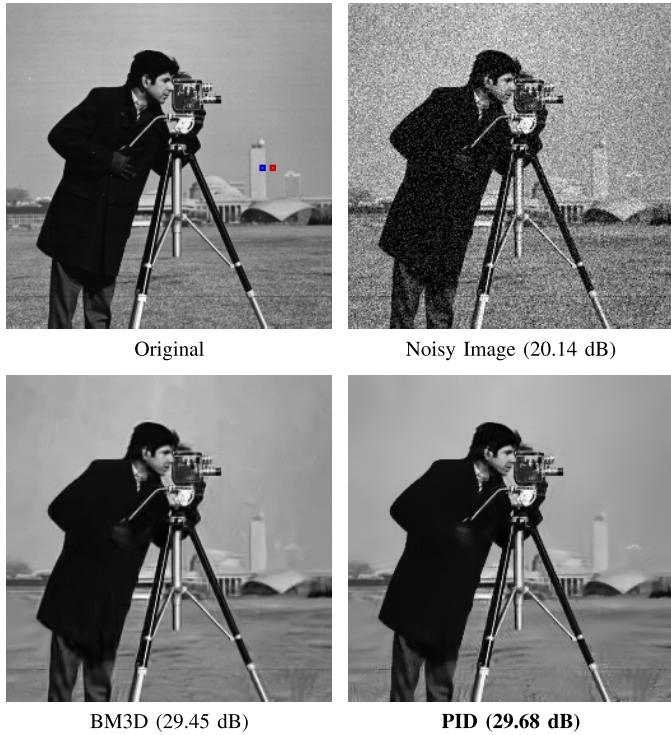


Fig. 7. Comparison of denoised *Cameraman* image with noise sigma $\sigma = 25$. PID produces homogeneous sky regions without objectionable artifacts. In the original image, we marked the two pixels which are considered in Fig. 6.

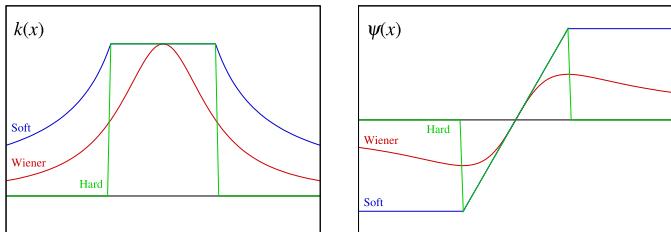


Fig. 8. Flipping wavelet shrinkage functions upside down allows reinterpretation as robust noise estimators. The left plot shows the weight functions $k(x)$ and the right plot shows the corresponding influence functions $\psi(x) = x k(x)$. Wiener filtering corresponds to the Lorentzian estimator and has poor outlier rejection. Soft thresholding corresponds to the Huber estimator, which is not redescending. Only hard thresholding has strong outlier rejection.

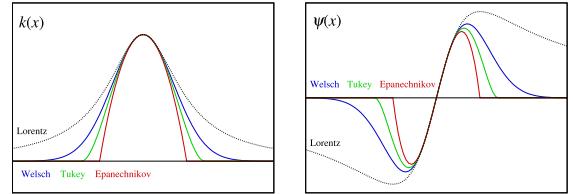


Fig. 9. Examples of redescending M -estimators. The left plot shows the weight functions $k(x)$ and the right plot shows the corresponding influence functions $\psi(x) = x k(x)$. The Lorentz estimator has the largest tail and does not sufficiently discard outliers. The Welsch estimator has a Gaussian as its weight function. It has a steep descent while still being infinite in support. The weight and influence functions of Tukey and Epanechnikov both have finite supports, rejecting outliers completely. The Wiener filter (Lorentzian estimator) is shown for reference.

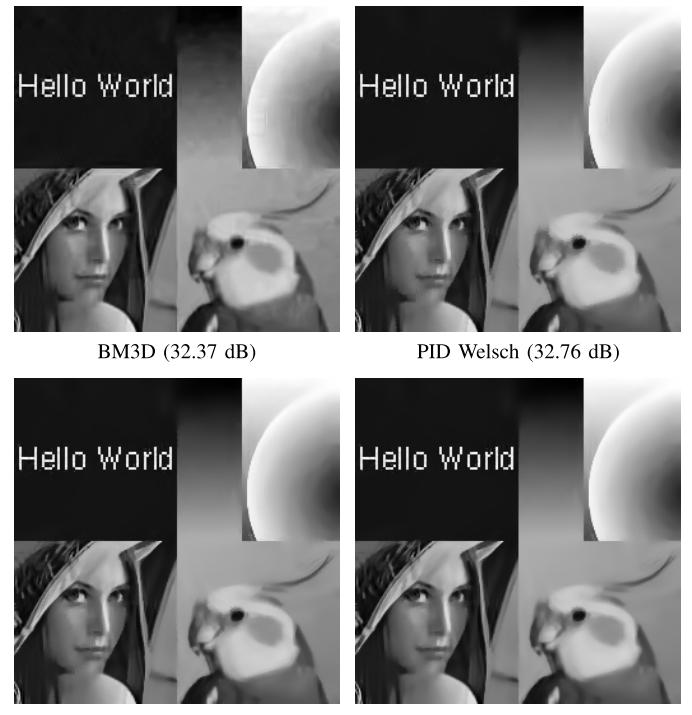


Fig. 10. Comparison of denoised *Montage* image using different redescending M -estimators for the range kernel. The noise sigma is $\sigma = 25$. The choice of redescending M -estimator has impact numerically, but visually, the images look similar.

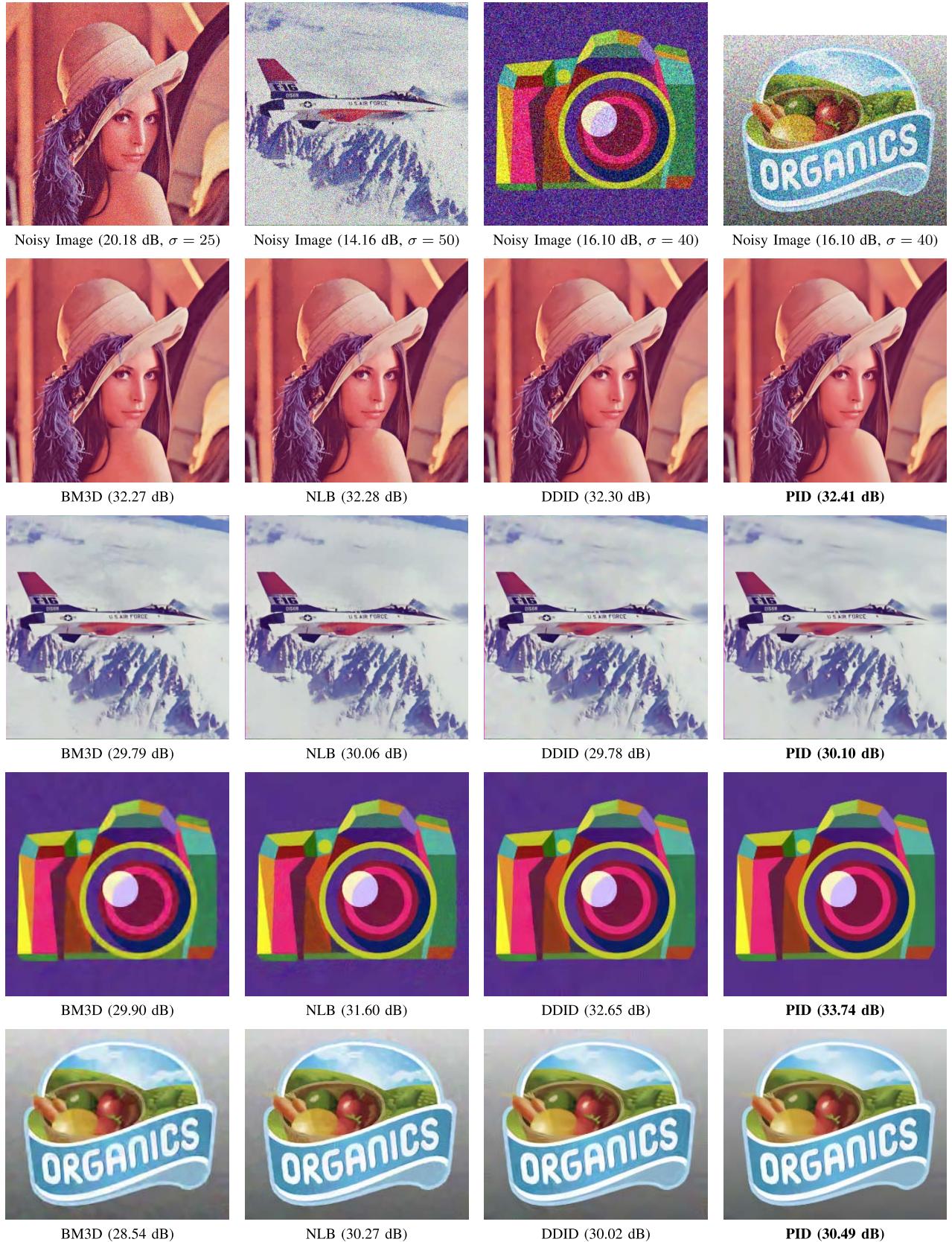


Fig. 11. Comparison of denoising natural and synthetic images. For natural images, PID produces smooth results for skin and homogeneous regions. For synthetic images, PID reconstructs clean contours and gradients.

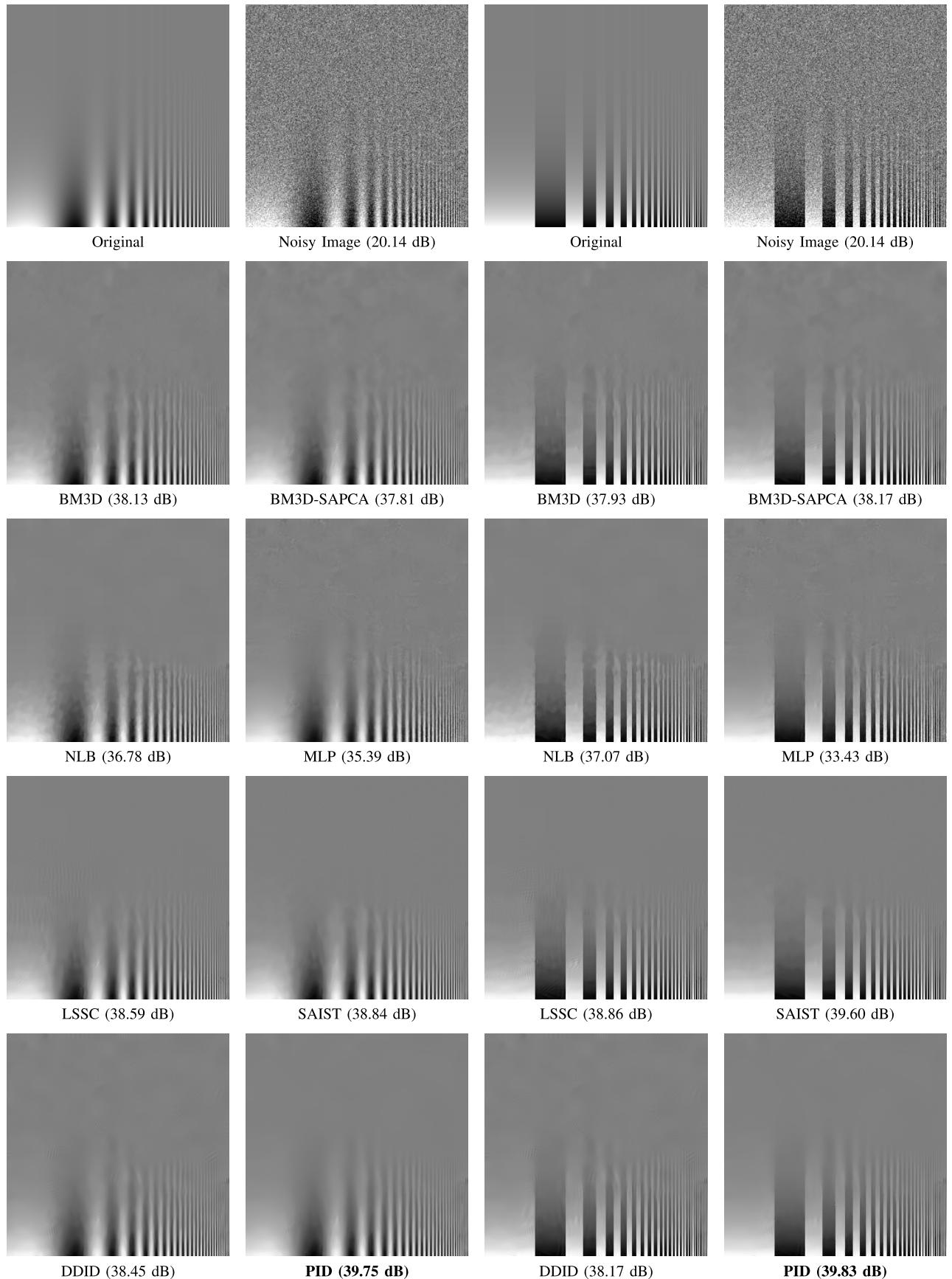


Fig. 12. Comparison of denoised CSF charts. Noise sigma is $\sigma = 25$. This table exposes characteristic artifacts of each method. Most methods exhibit low-frequency noise; NLB and SAIST additionally suffer from loss of contrast. The PID results are nearly artifact-free.

TABLE II

NUMERICAL COMPARISON FOR GRayscale AND COLOR IMAGES. THE TOP HALF CONTAINS PSNR VALUES, THE BOTTOM HALF SSIM

Grayscale $\sigma = 10$	Barbara	Boats	Cameraman	Couple	Finger Print	Hill	House	Lena	Man	Montage	Pepper
PID	34.55	33.77	34.14	33.91	31.78	33.60	36.60	35.81	34.03	37.43	34.67
DDID	34.67	33.74	34.05	33.88	31.84	33.56	36.50	35.81	34.02	37.51	34.58
SAPCA	35.10	34.10	34.59	34.17	32.64	33.83	37.01	36.07	34.25	37.85	34.94
SAIST	35.19	33.93	34.23	34.00	32.68	33.70	36.81	35.85	34.13	37.19	34.79
BM3D	34.98	33.92	34.18	34.04	32.46	33.62	36.71	35.93	33.98	37.35	34.68
MLP	34.07	33.81	34.18	33.91	32.57	33.59	35.98	35.85	34.11	36.51	34.72
LSSC	34.99	34.03	34.24	34.01	32.57	33.67	36.95	35.85	34.10	37.26	34.80
NLB	34.82	33.91	34.43	34.00	32.43	33.74	36.33	35.78	34.16	37.28	34.80
Grayscale $\sigma = 25$											
PID	30.56	29.80	29.68	29.65	27.15	29.77	32.84	32.12	29.68	32.76	30.37
DDID	30.80	29.79	29.47	29.56	27.32	29.71	32.66	32.14	29.62	32.61	30.29
SAPCA	31.00	30.03	29.81	29.82	27.81	29.96	32.96	32.23	29.81	32.97	30.43
SAIST	31.23	29.97	29.41	29.74	27.93	29.90	33.17	32.26	29.75	32.35	30.40
BM3D	30.72	29.91	29.45	29.72	27.70	29.85	32.86	32.08	29.62	32.37	30.16
MLP	29.55	29.97	29.61	29.74	27.65	29.88	32.57	32.26	29.89	32.04	30.31
LSSC	30.49	29.90	29.50	29.67	27.62	29.83	33.13	31.86	29.70	32.25	30.23
NLB	30.24	29.67	29.44	29.37	27.53	29.62	32.39	31.80	29.64	31.95	30.06
Grayscale $\sigma = 40$											
PID	28.38	27.71	27.60	27.40	24.98	27.92	30.76	30.14	27.66	30.25	28.10
DDID	28.51	27.65	27.32	27.30	25.04	27.83	30.41	30.07	27.60	29.82	27.94
SAPCA	28.68	27.92	27.57	27.58	25.54	28.08	30.75	30.10	27.83	30.02	28.10
SAIST	28.62	27.62	27.29	27.33	25.55	27.87	31.38	30.03	27.58	29.50	27.94
BM3D	27.99	27.74	27.18	27.48	25.30	27.99	30.65	29.86	27.65	29.52	27.70
MLP	n/a										
LSSC	28.17	27.77	27.34	27.41	25.30	28.00	31.10	29.91	27.64	29.43	27.86
NLB	28.05	27.39	27.13	27.18	25.39	27.76	30.25	29.83	27.52	29.10	27.66
Color $\sigma = 25$											
Baboon	F-16	House	Kodak 1	Kodak 2	Kodak 3	Kodak 12	Lake	Lena	Pepper	Tiffany	
PID	26.12	33.02	32.90	29.18	32.40	34.70	33.55	28.93	32.41	31.36	32.61
DDID	26.17	32.88	32.69	29.09	32.29	34.55	33.46	28.85	32.30	31.25	32.49
NLB	26.54	33.07	32.69	29.31	32.31	34.55	33.41	29.13	32.27	31.25	32.47
BM3D	25.95	32.78	33.03	29.13	32.44	34.54	33.76	28.68	32.27	31.20	32.23
Color $\sigma = 50$											
PID	23.41	30.10	30.54	25.86	29.81	31.39	30.82	26.57	30.07	29.34	30.12
DDID	23.31	29.78	29.99	25.75	29.61	31.09	30.53	26.41	29.82	29.13	29.85
NLB	23.63	30.06	30.38	26.03	29.45	31.08	30.50	26.71	29.85	29.09	29.61
BM3D	23.15	29.79	30.47	25.86	29.84	31.34	30.98	26.28	29.88	28.93	29.83
Grayscale $\sigma = 10$											
PID	0.9113	0.8389	0.9001	0.8715	0.9632	0.8451	0.8850	0.8687	0.8756	0.9539	0.8949
DDID	0.9116	0.8360	0.8933	0.8682	0.9641	0.8410	0.8825	0.8671	0.8729	0.9512	0.8900
SAPCA	0.9154	0.8452	0.9002	0.8774	0.9682	0.8513	0.9007	0.8712	0.8783	0.9545	0.8955
SAIST	0.9151	0.8400	0.9009	0.8729	0.9684	0.8488	0.8856	0.8711	0.8773	0.9539	0.8960
BM3D	0.9140	0.8387	0.8965	0.8739	0.9668	0.8430	0.8791	0.8690	0.8721	0.9529	0.8952
MLP	0.9037	0.8354	0.8953	0.8698	0.9676	0.8437	0.8498	0.8654	0.8739	0.9438	0.8938
LSSC	0.9118	0.8455	0.8952	0.8746	0.9675	0.8479	0.8891	0.8670	0.8748	0.9494	0.8940
NLB	0.9089	0.8375	0.8980	0.8712	0.9670	0.8479	0.8571	0.8639	0.8751	0.9465	0.8937
Grayscale $\sigma = 25$											
PID	0.8521	0.7177	0.8098	0.7472	0.8930	0.6889	0.7876	0.7999	0.7371	0.9113	0.8247
DDID	0.8499	0.7180	0.7909	0.7447	0.8949	0.6898	0.7790	0.7956	0.7346	0.8935	0.8126
SAPCA	0.8534	0.7320	0.8148	0.7623	0.9078	0.7099	0.7882	0.7998	0.7507	0.9015	0.8193
SAIST	0.8591	0.7232	0.7957	0.7507	0.9080	0.6985	0.7866	0.7998	0.7421	0.9080	0.8262
BM3D	0.8454	0.7295	0.8014	0.7619	0.9053	0.7057	0.7854	0.7937	0.7433	0.8963	0.8178
MLP	0.8205	0.7322	0.8154	0.7598	0.9068	0.7072	0.7814	0.8007	0.7536	0.8974	0.8233
LSSC	0.8372	0.7237	0.7989	0.7584	0.9032	0.7076	0.7946	0.7877	0.7450	0.8889	0.8097
NLB	0.8335	0.7160	0.8093	0.7332	0.9055	0.6917	0.7701	0.7835	0.7439	0.8778	0.8091
Grayscale $\sigma = 40$											
PID	0.7917	0.6370	0.7542	0.6547	0.8280	0.6006	0.7521	0.7543	0.6495	0.8727	0.7723
DDID	0.7895	0.6364	0.7264	0.6503	0.8303	0.5996	0.7307	0.7428	0.6454	0.8376	0.7500
SAPCA	0.7950	0.6589	0.7526	0.6755	0.8543	0.6254	0.7460	0.7483	0.6682	0.8491	0.7620
SAIST	0.7882	0.6333	0.7386	0.6539	0.8455	0.6005	0.7585	0.7445	0.6484	0.8614	0.7671
BM3D	0.7688	0.6542	0.7419	0.6740	0.8485	0.6233	0.7406	0.7353	0.6612	0.8372	0.7543
MLP	n/a										
LSSC	0.7698	0.6461	0.7537	0.6666	0.8444	0.6187	0.7525	0.7414	0.6593	0.8470	0.7556
NLB	0.7670	0.6187	0.7326	0.6449	0.8514	0.5990	0.7330	0.7365	0.6459	0.8210	0.7405
Color $\sigma = 25$											
Baboon	F-16	House	Kodak 1	Kodak 2	Kodak 3	Kodak 12	Lake	Lena	Pepper	Tiffany	
PID	0.8121	0.8676	0.8289	0.8295	0.7516	0.8753	0.7923	0.7753	0.8364	0.7821	0.8068
DDID	0.8113	0.8644	0.8356	0.8287	0.7559	0.8740	0.7969	0.7767	0.8359	0.7797	0.8066
NLB	0.8257	0.8675	0.8104	0.8320	0.7592	0.8678	0.7859	0.7841	0.8319	0.7813	0.8079
BM3D	0.8056	0.8679	0.8307	0.8191	0.7653	0.8714	0.8089	0.7805	0.8371	0.7850	0.8035
Color $\sigma = 50$											
PID	0.6450	0.8103	0.7715	0.6473	0.6568	0.7951	0.6986	0.6930	0.7782	0.7408	0.7336
DDID	0.6522	0.7981	0.7511	0.6595	0.6547	0.7884	0.6949	0.6890	0.7704	0.7295	0.7251
NLB	0.6739	0.8111	0.7675	0.6747	0.6454	0.7798	0.6809	0.7075	0.7690	0.7348	0.7217
BM3D	0.6391	0.8076	0.7694	0.6585	0.6718	0.7917	0.7160	0.6998	0.7748	0.7341	0.7310

well for natural images. The last two examples contrast the qualities between other methods and our method for synthetic images. With PID, gradients are smoother, edges are sharper, tips are clearer, and text is crisper.

Numerically, PID is more competitive than DDID. Table II summarizes results for grayscale and color images. The

grayscale images are the same set tested by BM3D and mostly originate from the Signal and Image Processing Institute (SIPI) database. Some of the SIPI/BM3D images have synthetic character with large amount of homogenous regions like sky or gradients. These are *Cameraman*, *House*, *Lena*, *Montage*, and *Pepper*. For these images, we observe most

of the gain using PID. Other images have more “natural” character and there is not much to improve. Highly stochastic images like *Finger Print* are hard to improve. Since PID excels at denoising highly correlated images, it is not a surprise that it performs well on color images. Since color channels are highly correlated, signals and noise are better separated. We compare PID against the subset of the state-of-the-art methods which support denoising of color images, which are BM3D, DDID, and NLB. For most images, especially for high-noise scenarios, PID exceeds BM3D and DDID.

B. Artifacts Study

Artifacts are sometimes hard to spot. When looking at natural images, artifacts may actually increase the perceived realism, due to structures in the image that are recognized as detail. Examples are the clouds in the *F16* image of Fig. 11. Careful observation shows that such structures are not existent in the original image and are in fact residual noise. In synthetic images these artifacts are more apparent since we have an idealized reference image in mind.

Since most image denoising methods are biased towards natural images, where the PSNR results are close to the theoretical limit [2], it is desirable to have synthetic images which can be evaluated numerically and visually. We found that the Campbell-Robson contrast sensitivity function (CSF) charts are suitable candidates. CSF charts are synthetic images used for evaluating human perception of contrast as a function of frequencies. They contain therefore a continuous range of amplitudes and frequencies in a single image. Fig. 12 demonstrates the effectiveness of CSF charts for evaluating image denoising methods. The table shows that most denoising methods have difficulties in homogeneous regions. Also, the denoised CSF charts reveal characteristic artifacts for every method. The CSF images are generated using the CSF chart function described by Pelli [21] as

$$f(x, y) = \frac{1}{2} (1 + 10^{-y} \sin 10^x). \quad (10)$$

We bound the signal bandwidth by Nyquist frequency by constraining

$$x \in \left[0, \frac{W(\pi w)}{\log 10} \right],$$

where $W(\cdot)$ is the Lambert W function and w the width of the image in pixels. We allow the contrast to decrease to 10^{-3} with $y \in [0, 3]$. For our version with steps, we use

$$f(x, y) = \frac{1}{2} (1 + 10^{-y} \operatorname{sgn}(\sin 10^x)). \quad (11)$$

C. Performance

Compared to DDID, PID uses more iterations. Every pixel requires a Fast Fourier Transform (FFT) per iteration, which cannot be optimized across neighbors due to potentially very different bilateral masks. Our focus in this work is to emphasize on the quality and simplicity of the denoising process.

We can foresee some performance improvements in the near future. For example, it would be possible to adaptively grow the window radius to follow the growing spatial kernel. Our method is straightforward to parallelize. Denoising a 256×256 grayscale image with our MATLAB implementation using 8 cores on an Intel Xeon 2.67 GHz takes 100 seconds. Our CUDA implementation using a NVIDIA GTX Titan reduces the time to 2.5 seconds.

VI. DISCUSSION

Our robust noise estimation paradigm unifies spatial and frequency domain denoising. Durand and Dorsey have already made the connection between bilateral filter and robust estimation [14]. The bilateral filter is robust in three dimensions, two spatial and one range dimension for the spatial domain. Our estimator adds another range dimension for the frequency domain. Since both domains induce bias, our robust noise estimator protects from bias in both domains.

The original bilateral filter is defined using the Welsch estimator for both spatial and range dimensions. Durand and Dorsey considered alternative robust estimators like the Lorentz and Tukey estimator. To add even more flexibility, it is possible to pick different robust estimators for every dimension individually.

For example, we can consider alternative weight functions for the frequency and, more generally, wavelet domain. In the wavelet domain, denoising is usually performed by shrinkage of the wavelet coefficients. The idea is to preserve the signal and to discard the noise. In our denoising framework, we consider signal as the outlier and noise as the inlier being estimated. We denoise by subtracting the estimated noise, rather than estimating the denoised signal directly. We take an alternative perspective by considering wavelet shrinkage as robust noise estimation. Fig. 8 reveals typical shrinkage operators as robust estimators of noise. The most popular wavelet shrinkage method is the mean squared error (MSE) minimizing Wiener filter. However, we can also understand the Wiener filter as a robust noise estimator. If we subtract the Wiener filter from 1 as

$$K_{p,f} = 1 - \frac{|D_{p,f}|^2}{|D_{p,f}|^2 + \sigma_{p,f}^2} = \frac{1}{|D_{p,f}|^2 / \sigma_{p,f}^2 + 1}, \quad (12)$$

we recognize the Lorentzian redescending M-estimator. While its influence function is redescending, the Lorentzian has a heavy tail, failing to discard strong signals as outliers, leading to ringing artifacts. The Wiener filter is therefore a poor choice for outlier rejection. The same analysis can be made for thresholding. Soft thresholding is the Huber estimator in disguise, which is not redescending, making it unusable for robust noise estimation. Hard thresholding, on the other hand, is a redescending M-estimator with strong outlier rejection.

Since kernels with heavier tails than the Gaussian are insufficient to remove outliers, we choose instead robust estimators with stronger outlier rejection. The most typical redescending M-estimators are depicted in Fig. 9. In Table I, we compare the PSNRs using the different robust kernels for the range kernel k_r . All other kernels are Gaussians. A visual comparison

is made in Fig. 10. For simplicity, we settled on using the Welsch/Gaussian robust estimator.

VII. CONCLUSIONS

We presented image denoising as a physical process, which can be summarized in three points. First, we perform a gradient descent by progressively estimating noise differentials and subtracting them iteratively from the noisy image. Second, the noise differentials are estimated using robust kernels in two spatial domains, one spatial range domain and one frequency range domain. Third, the kernel scale parameters are modified according to an annealing schedule.

Our approach using robust estimators unifies spatial and wavelet domain methods. We considered denoising as robust noise estimation in multiple dimensions and domains. In each dimension and domain, the noise estimation is protected from bias by using redescending M-estimators. This perspective allowed us to reinterpret wavelet shrinkage as robust noise estimation.

We also connected image denoising to statistical mechanics. Similar to deterministic annealing, we change the shape of the robust noise estimator over time. The scale parameter of the range kernel corresponds to the falling temperature in deterministic and simulated annealing. This annealing process allows us to find near-optimal solutions.

In contrast to current state-of-the-art denoising methods, our algorithm is short. Despite its simplicity, the algorithm delivers high-quality results, outperforming other methods in denoising synthetic images. Considering that many methods already work well enough for natural images, the new challenge is in denoising synthetic images.

State-of-the-art denoising methods are currently too slow for integration into consumer products. Since we focused on quality and simplicity rather than performance, there remains a terrain to explore acceleration of our denoising process.

Finally, we believe that our contribution is not limited to image denoising. An avenue of investigation would be to find out how our denoising approach impacts related problems like artifact removal, superresolution, and hole filling. Our denoising formulation is agnostic of dimensionality of the signal. We therefore expect that our contribution to be of interest to the signal processing community at large and to many domain specific applications.

ACKNOWLEDGMENTS

We thank Charis Tsevis for allowing us to use his artistic rendering of a Nikon camera in Fig. 11. The *Organics* logo in Fig. 11 was designed by *Logo Open Stock* [22].

REFERENCES

- [1] K. Dabov, A. Foi, V. 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.
- [2] A. Levin and B. Nadler, "Natural image denoising: Optimality and inherent bounds," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 2833–2840.
- [3] P. Chatterjee and P. Milanfar, "Is denoising dead?," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 895–911, Apr. 2010.
- [4] A. Levin, B. Nadler, F. Durand, and W. T. Freeman, "Patch complexity, finite pixel correlations and optimal denoising," in *Proc. 12th ECCV*, Oct. 2012, pp. 73–86.
- [5] C. Knaus and M. Zwicker, "Dual-domain image denoising," in *Proc. 20th IEEE ICIP*, Sep. 2013, pp. 440–444.
- [6] S. Z. Li, "Robustizing robust M-estimation using deterministic annealing," *Pattern Recognit.*, vol. 29, no. 1, pp. 159–166, Jan. 1996.
- [7] R. Frühwirth and W. Waltenberger, "Redescending m-estimators and deterministic annealing, with applications to robust regression and tail index estimation," *Austrian J. Statist.*, vol. 37, nos. 3–4, pp. 301–317, 2008.
- [8] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "BM3D image denoising with shape-adaptive principal component analysis," in *Proc. SPARS*, 2009.
- [9] M. Lebrun, A. Buades, and J. Morel, "A nonlocal Bayesian image denoising algorithm," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1665–1688, 2013.
- [10] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 2272–2279.
- [11] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, Feb. 2013.
- [12] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 2392–2399.
- [13] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, Jan. 1998, pp. 839–846.
- [14] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 257–266, 2002.
- [15] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.
- [16] V. Černý, "Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm," *J. Optim. Theory Appl.*, vol. 45, no. 1, pp. 41–51, Jan. 1985.
- [17] P. Carnevali, L. Coletti, and S. Patarnello, "Image processing by simulated annealing," *IBM J. Res. Develop.*, vol. 29, no. 6, pp. 569–579, Nov. 1985.
- [18] Q. Lu and T. Jiang, "A new Bayesian approach to image denoising with a combination of MRFs and pixon method," in *Proc. 15th Int. Conf. Pattern Recognit.*, vol. 3, 2000, pp. 726–729.
- [19] L. Yan, L. Wang, and K.-H. Yap, "A noisy chaotic neural network approach to image denoising," in *Proc. ICIP*, vol. 2, Oct. 2004, pp. 1229–1232.
- [20] L. Wang, L. Yan, and K.-H. Yap, "Image denoising using stochastic chaotic simulated annealing," in *Proc. Intell. Multimedia Process. Soft Comput.*, 2005, pp. 415–429.
- [21] D. G. Pelli, "Programming in postscript: Imaging on paper from a mathematical description," *BYTE*, vol. 12, no. 5, pp. 185–202, 1987.
- [22] (2013, Aug. 19). *Logo Open Stock* [Online]. Available: <http://www.logopenstock.com>

Claude Knaus received the Ph.D. degree in computer science from the University of Bern, Bern, Switzerland. His research interests include rendering and image denoising. Before that, he spent 10 years in the industry, including research and development positions at Silicon Graphics Inc., Sunnyvale, CA, USA, and Object Technology International, Ottawa, ON, Canada, and IBM, Ottawa. He was also involved with the Khronos Group in the specification of OpenGL ES and OpenGL SC. He received the degree in computer science from ETH Zurich, Zurich, Switzerland, in 1998.

Matthias Zwicker has been a Professor with the University of Bern, Bern, Switzerland, where he has been the Head of the Computer Graphics Group with the Institute for Computer Science and Applied Mathematics since 2008. He was an Assistant Professor at the University of California at San Diego, La Jolla, CA, USA, from 2006 to 2008. He has served as a Papers Co-Chair and Conference Chair of the IEEE/Eurographics Symposium on Point-Based Graphics, and as a Papers Co-Chair for Eurographics 2010. He has been a Program Committee Member for various conferences, including ACM SIGGRAPH and Eurographics, and an Associate Editor of the Computer Graphics Forum and the *Journal of the Eurographics Association*.