Real image denoising with a locally-adaptive bitonic filter

Graham Treece

Technical Report ENG-TR.006
ISSN 2633-6839
September 2021

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

Corresponding e-mail: gmt11@cam.ac.uk

Abstract

The bitonic filter is a non-learning-based filter for removing noise from signals, following mathematical morphology (ranking) approaches, relying on a novel presumption in which the signal is postulated to be locally bitonic (having only one minimum or maximum) over some domain of finite extent. It is here developed specifically for image noise so that the domain is locally-adaptive, leading to significant improvements in noise reduction performance at no cost to processing times. The new bitonic filter performs better than the block-matching 3D filter for high levels of additive white Gaussian noise, and over all noise levels for two public data sets containing real image noise. This is despite an additional adjustment to the block-matching filter for real image noise, which leads to significantly better performance than has previously been cited on these data sets. The new bitonic filter has a signal-to-noise ratio only 2.4 dB lower than the best learning-based techniques when they are optimally trained. The performance gap is closed completely when these techniques are trained on data sets not directly related to the benchmark data. This demonstrates what can be achieved with a predictable, explainable, entirely local technique, which makes no assumptions of repeating patterns either within an image or across images, and hence creates residual images which are well behaved even in very high noise. Since the filter does not require training, it can still be used in situations where training is either difficult or inappropriate.

1 Introduction

The removal of noise from images has been the subject of considerable research for several decades, with the focus recently shifting almost entirely to learning-based techniques (Gu and Timofte, 2019; Tian et al., 2020). All approaches make assumptions about the properties of the noise and the signal, whether explicitly modelled or learned, and the specificity of these assumptions affects the performance and the generality of the techniques. Learning-based methods currently have the best performance, provided they are carefully trained on appropriately similar data, but explicit methods can exhibit greater generality in performing better on new data sets with different noise features (Abdelhamed et al., 2018; Anaya and Barbu, 2018; Plotz and Roth, 2017).

The bitonic filter is a recently-developed non-learning-based technique, built on mathematical morphology (ranking) rather than linear convolutions. It relies on a novel noise model, in which the signal is postulated to be locally bitonic (having only one minimum or maximum) over some domain of finite extent. This domain was initially a fixed shape (Treece, 2016) then took a structurally-varying form (Treece, 2019). The resulting filter outperformed many other traditional filters, but only competed with block-matching (BM3D — usually the standard non-learning based filter against which others are now compared) (Dabov et al., 2007) at very high noise levels.

It was suggested that further improvements may be possible from the use of more complex domains (mask shapes) (Treece, 2019), and this is the primary focus of this paper. However, all the previous results were based on adding simulated noise to noise-free images, whereas typical images from modern smartphones or larger lens cameras exhibit quite different noise properties, and have consequently shown quite different performances across noise-reduction techniques (Abdelhamed et al., 2018; Plotz and Roth, 2017). Hence this paper additionally investigates performance on two recently developed public datasets which are based on images with real noise.

Pursuing a non-learning-based filter when the learning-based alternatives typically perform better if they are trained appropriately is perhaps questionable, but there are good reasons for doing so. Firstly, because the filter is derived entirely from local noise assumptions, which are neither based on any sort of pattern matching within an image nor training across multiple images, it is impossible to generate false local image features which are actually from another part of the image or from another image entirely. Hence the residual noise-reduced image will always be both predictable and explainable, even for very noisy situations and in as yet untested scenarios. Secondly, the filter is equally applicable in any situation in which training is either difficult or inappropriate and, as suggested by the later results, the performance is also likely to have greater generality given it is not trained on a specific situation. Thirdly, on real images the new filter narrows the performance gap between traditional and learning-based methods, and reveals what is possible without making any assumptions about repeating patterns, hence also moderating slightly what is gained from specific training.

In demonstrating improved performance over both BM3D and the original instantiations of the bitonic filter, particularly on real image noise, it has also become clear that the performance of BM3D has been somewhat under-reported in this scenario, with some fairly simple changes considerably improving on previously published results.

2 Methods

The existing bitonic filter is briefly summarised in Sections 2.1 and 2.2, since it is not well known, including some minor changes to the multi-resolution framework. The locally-adaptive domain is developed in Section 2.3 and application to images with real noise is discussed in Section 2.4. Consideration is then given to the key threshold parameter in Section 2.5 and to efficient implementation of the locally-adaptive ranking operations in Section 2.6.

2.1 Bitonic filter

The constant-shape bitonic filter was analysed in detail in (Treece, 2016) and extended to a structurally-varying version in (Treece, 2019). It consists of a robust opening $O_{w,c}$ and closing $C_{w,c}$ of a signal $I(\mathbf{x})$ which varies with location $\mathbf{x}$:

$$R_{w,c}(I)(\mathbf{x}) = \underset{\mathbf{y}\in w(\mathbf{x})}{c^{\text{th}}\text{centile}} \{I(\mathbf{x}+\mathbf{y})\} \quad (1)$$

$$R^{-1}_{w,c}(I)(\mathbf{x}) = \underset{-\mathbf{y}\in w(\mathbf{x}+\mathbf{y})}{c^{\text{th}}\text{centile}} \{I(\mathbf{x}+\mathbf{y})\} \quad (2)$$

$$O_{w,c} = R^{-1}_{w,100-c}(R_{w,c}(I)) \quad (3)$$

$$C_{w,c} = R^{-1}_{w,c}(R_{w,100-c}(I)) \quad (4)$$

where $R_{w,c}$ is a rank filter, and $\mathbf{y}$ is a vector offset to a location close to $\mathbf{x}$, within a filter region w (or mask in 2D). This ranks (sorts) I over w and returns the intensity corresponding to the chosen centile c from the sorted list. All $|w|$ mask elements are constrained to be within a square region of size $l \times l$. The centile c is set to 8% throughout this paper: setting $c = 100\%$ in eq. (1) would return the maximum value (known as a dilation) and $c = 0\%$ the minimum (known as an erosion), but in practice a small non-zero centile value as in Kass and Solomon (2010) gives more robust results in the presence of noise. If w is a simple shape and constant in $\mathbf{x}$, the reverse rank filter $R^{-1}_{w,c}$ is the same as $R_{w,c}$, i.e. w at $\mathbf{x}$ must contain $\mathbf{y}$ for inclusion of data from $\mathbf{x}+\mathbf{y}$. For $R^{-1}_{w,c}$ with varying masks, what matters is whether the masks used in the forward rank at surrounding locations $\mathbf{x}+\mathbf{y}$ overlap with $\mathbf{x}$, i.e. contain $-\mathbf{y}$.

Opening and closing operations are not self-dual (symmetric in data value), and do not preserve the mean signal intensity. To correct this defect, the operations are weighted, by considering their difference from the original signal. This difference is smoothed with an anisotropic Gaussian filter $G_{\sigma,\alpha}$.

$$\epsilon_O = |G_{\sigma,\alpha}\left(I(\mathbf{x}) - O_{w,c}(\mathbf{x})\right)| \quad (5)$$

$$\epsilon_C = |G_{\sigma,\alpha}\left(C_{w,c}(\mathbf{x}) - I(\mathbf{x})\right)| \quad (6)$$

$$b_{w,c} = \frac{\epsilon_O^m\left(C_{w,c} - \epsilon_C\right) + \epsilon_C^m\left(O_{w,c} + \epsilon_O\right)}{\epsilon_O^m + \epsilon_C^m} \quad (7)$$

where $b_{w,c}$ is the output of the bitonic filter, with $m = 3$ controlling the sharpness of the transition between $C_{w,c}$ and $O_{w,c}$. The opening and closing operations effectively detect bitonic signals, and hence ϵ_O and ϵ_C represent smoothed versions of the residual errors after removing such signals from the original. The result is to preserve bitonic signals, but reduce noise in all regions, including across signal edges.

The anisotropic filter $G_{\sigma,\alpha}$ is sensitive to the local direction $\phi(\mathbf{x})$ and degree of anisotropy $\gamma(\mathbf{x})$ in the image $I(\mathbf{x})$ which, following Van Vliet and Verbeek (1995), are derived from the eigenvalues of the structure tensor. $\gamma = 0$ signifies low anisotropy (no dominant direction) and $\gamma = 1$ signifies high anisotropy (local gradients in only one direction). $\phi(\mathbf{x})$ is the angle following the dominant features in the image, i.e. the direction in which the filter should be aligned. These can be used to construct a Gaussian-like filter which has a spatial range given by σ (here set to $0.21l$), and rotational asymmetry α (set to 0.6), controlling the extent to which the filter follows the dominant image direction and anisotropy:

$$d_{\mathbf{y}}(\mathbf{x}) = |\mathbf{y}\sin(\angle(\mathbf{y}) - \phi(\mathbf{x}))| \quad (8)$$

$$\Omega_{\mathbf{y}}(\mathbf{x}) = \frac{e^{-\frac{|\mathbf{y}|^2}{2\sigma^2}}}{\left(\frac{d_{\mathbf{y}}(\mathbf{x})\gamma(\mathbf{x})^2}{\alpha^2}+1\right)\left(\frac{d_{\mathbf{y}}(\mathbf{x}+\mathbf{y})\gamma(\mathbf{x}+\mathbf{y})^2}{\alpha^2}+1\right)} \quad (9)$$

$$G_{\sigma,\alpha}(I)(\mathbf{x}) = \frac{\sum_{\mathbf{y}\in w_l}\Omega_{\mathbf{y}}(\mathbf{x})I(\mathbf{x}+\mathbf{y})}{\sum_{\mathbf{y}\in w_l}\Omega_{\mathbf{y}}(\mathbf{x})} \quad (10)$$

In this case $\mathbf{y}$ is restricted to a rectangular mask w_l of size $l \times l$. It is also possible to introduce a threshold on $|I(\mathbf{x}) - I(\mathbf{x}+\mathbf{y})|$ above which $\Omega_{\mathbf{y}}(\mathbf{x}) = \Omega_{\mathbf{x}}(\mathbf{y}) = 0$.

For colour images, $\phi(\mathbf{x})$ and $\gamma(\mathbf{x})$ are calculated from the noisy grey-scale image and used to create the smoothed grey-scale data for local mask definition as in Section 2.3. Subsequent operations are performed independently for each colour channel, but application of Eqs. (5) and (6) all use the same $\phi(\mathbf{x})$ and $\gamma(\mathbf{x})$ to ensure there is no colour separation due to smoothing in different directions.

2.2 Multi-resolution framework

As explained in (Treece, 2019), the bitonic filter is set in a multi-resolution framework by use of a threshold t_n which is lowered with each level n, and applied to a reduced, processed image. Figure 1 details the approach adopted in this paper, which follows this with only minor changes. Reduction of the image to quarter size is achieved with a Catmull-Rom spline (Catmull and Rom, 1974) restriction operator, as is typical in the multi-grid framework (Briggs et al., 2000). Expansion of the image is by the adjunct prolongation operator. These operations, denoted by diagonal arrows in Fig. 1, are not lossless: to account for this, the prolongated, restricted image, without any further processing, is subtracted from the original before adding in the lower level results.

The threshold t_1 for the first level is set to the noise range t and subsequent thresholds t_n are calculated from the expected reduction in noise due to the bitonic filter applied at the previous level. For the first level this will be inversely proportional to filter length l, since the noise variance reduces approximately with the size of the mask. However, and in contrast to previous work, for subsequent levels the reduction is by a factor of two:

$$t_n = \begin{cases} \frac{t_1}{1.2l} & n = 2 \\ \frac{t_{n-1}}{2} & n > 2 \end{cases} \quad (11)$$

At these lower levels the previous filtering has introduced significant correlation between neighbouring image pixels. Hence the noise reduction from further filtering is dominated by the increase in filter area, rather than the number of image pixels included in the filter at that level. Since the images are reduced to quarter size, but the filter size is kept constant, this results in a four-fold reduction in noise variance (two-fold in noise range).

Once the lower levels have been processed, the results are prolongated and added back into the previous result at the higher level. Some additional noise reduction is achieved by repeated application of the bitonic at this level, but with t_{n+1}, i.e. the threshold from the lower level.

2.3 Locally-adaptive masks

In (Treece, 2019) the mask shape $w(\mathbf{x})$ was allowed to vary locally, by selection from a relatively small set of pre-defined elliptical masks, with different orientations and aspect ratios. This constraint was important since, without it, the mask shape could potentially adapt to the noise rather than the signal in an image and subsequently end up highlighting noise rather than removing it. For a more flexible local mask which can further adapt to the local signal shape, care is needed to find ways to exclude the effects of noise on this shape. The adaptibility is achieved by included mask elements which are within some data threshold of the current location $\mathbf{x}$; the resistance to noise is achieved by making use of several carefully designed thresholds across different image domains. This new filter is called Bitonic X (or MX for the multi-resolution version).

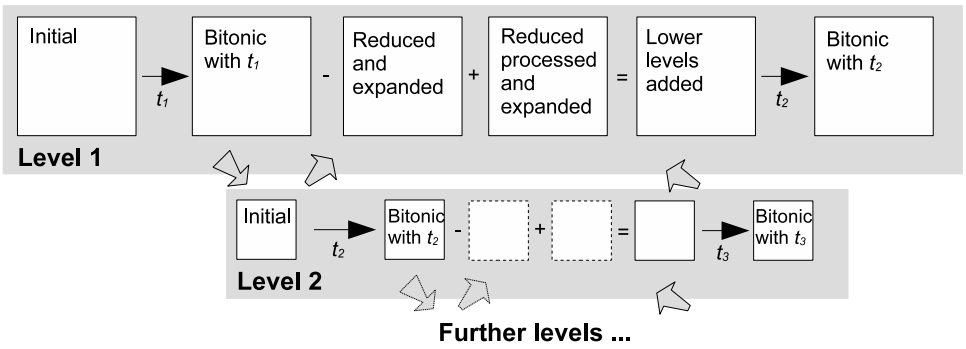


Figure 1: Multi-resolution implementation of the bitonic filter. At each lower level n, the filter operates on a quarter-size image, with a reduced threshold t_n, but with the same filter length l, so the effective filter range increases. t_n is reduced to account for the reduced noise level due to the bitonic filter previously applied at the higher level. Results from lower levels are expanded and added back in, taking into account the lossy nature of reduction (restriction) and expansion (prolongation). The bitonic filter is then re-applied at this level, but with the lower threshold from the next level down.

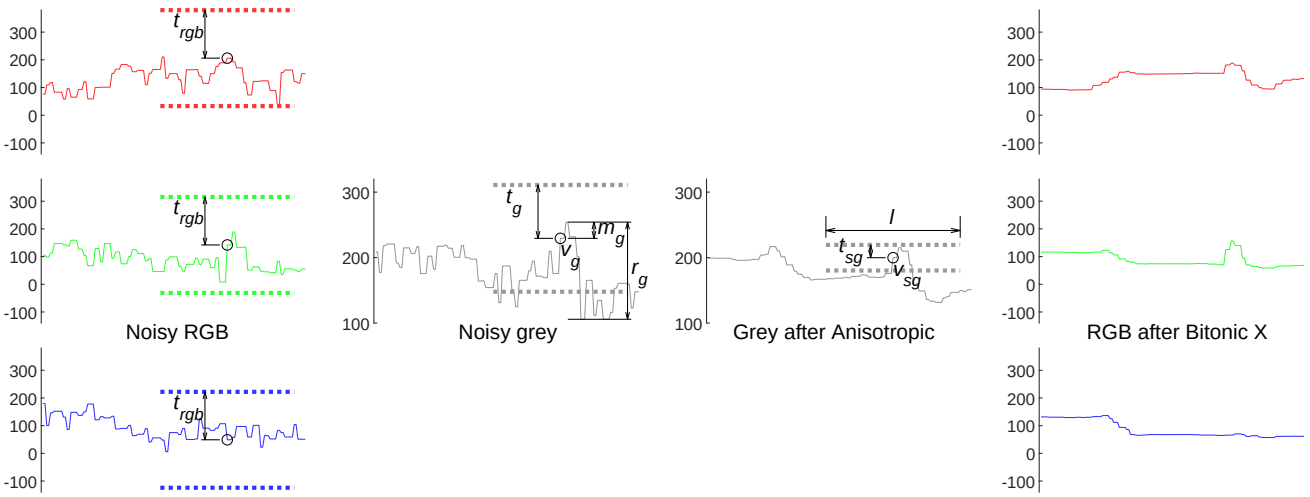


Figure 2: Use of thresholds in Bitonic X to define the locally-adaptive mask $w(\mathbf{x})$. The graphs show data values along the same line passing through an image. Thresholds t_{rgb}, t_g and t_{sg}, respectively, are applied to the noisy RGB, grey-scale, and smoothed grey-scale images. Mask elements are only used where the corresponding pixel is within this threshold of the centre pixel for all these domains. The thresholds are derived from an initial constant t, and local grey-scale statistics m_g and r_g. In this relatively high noise case, the mask is mostly determined by the smoothed grey-scale image. This is a one-dimensional example for simplicity: in reality the application is in two dimensions.

Figure 2 shows the image domains and defines the terms used in creating thresholds. The noisy RGB data (or other channels if operating in a different colour space) has the highest noise level, but the signal is otherwise unprocessed, and it can reveal colour boundaries which may not be present in the grey-scale image. The grey-scale data has somewhat reduced noise due to averaging over the raw colour channels, with no spatial blurring of the signal. The smoothed grey-scale data has a much lower noise level, but at the cost of some corruption of the signal. All three domains can potentially contribute to the selection of mask elements, but the properties and noise levels of each are different and so different thresholds t_{rgb}, t_g and t_{sg} need to be developed for each domain. These are based on an initial constant value t which is typically set to four times the standard deviation of the noise σ_n in each of the raw colour channels. An element is included in the local mask $w(\mathbf{x})$ only if it is within the relevant threshold for *all* the domains.

The initial threshold t is first adjusted by a factor f which depends on the local noisy grey-scale statistics:

$$f_{\text{low}} = \frac{2}{3}, \quad f_{\text{high}} = \frac{3}{2}, \quad r_{\text{low}} = t, \quad r_{\text{high}} = 1.6t \tag{12}$$

$$f' = \begin{cases} f_{\text{low}} & r_g > r_{\text{high}} \\ f_{\text{high}} & r_g < r_{\text{low}} \\ \frac{f_{\text{high}}(r_{\text{high}} - r_g) + f_{\text{low}}(r_g - r_{\text{low}})}{r_{\text{high}} - r_{\text{low}}} & \text{otherwise} \end{cases} \tag{13}$$

$$f = \begin{cases} 1 - \frac{m_g}{f_{\text{low}}}(1 - f') & (f' < 1)\,\&\,(m_g < f_{\text{low}}) \\ f' & \text{otherwise} \end{cases} \tag{14}$$

where m_g and r_g are derived from the local grey-scale statistics, as in Fig. 2 and f_{low}, f_{high}, r_{low} and r_{high} are constants.

The factor f' allows a range of thresholds from $f_{\text{low}}t$ to $f_{\text{high}}t$. The lower limit is appropriate when the noise level is low compared to the local range of data values. In this case the data comes from several, possibly overlapping, distributions and hence there is a greater chance of incorrectly including a mask element which is not from the same distribution as the current location $\mathbf{x}$. The higher limit is appropriate when the local data values are within the expected noise range, and hence there is a high chance that all the local data is from the same distribution. Equation (13) controls the transition between these two regions, which depends on the local range of the data r_g.

It is also possible that the data at $\mathbf{x}$ is sufficiently close to the local minimum or maximum, given the noise level, to indicate it is biased away from the mean. Equation (14) accounts for this by setting f from f' according to the distance from the nearest extremum, m_g, to allow for the greater range necessary to cover all data from this distribution. f is then used to scale the threshold t locally:

$$t_g = \frac{1}{\sqrt{3}} f t \tag{15}$$

$$t_{\text{rgb}} = \begin{cases} \sqrt{3}t & f \le 1 \\ \sqrt{3}ft & f > 1 \end{cases} \tag{16}$$

$$t'_{\text{sg}} = \begin{cases} \frac{2}{\sqrt{\pi}l} f t_g & \text{level } n = 1 \\ \frac{1}{2} f t_g & \text{otherwise} \end{cases} \tag{17}$$

where t_{rgb} and t_g are the thresholds applied to noisy RGB and grey-scale data, respectively, as in Fig 2. The grey-scale image is created by averaging the RGB values, so the noise is reduced by a factor of $\sqrt{3}$ compared to the noisy RGB data, hence t_g is lower than t. In contrast, t_{rgb} has an additional factor of $\sqrt{3}$, since there are three raw channels, each presumed to have independent noise. Without this factor, there would be a greater chance of a noisy RGB level incorrectly being detected within this threshold just because there are three channels to test.

The threshold for smoothed grey-scale data t'_{sg} is set according to the expected noise reduction from the anisotropic smoothing filter. As explained in Section 2.2, this reduction is proportional to the square root of the filter area for the first level, but is a factor of two for subsequent levels due to inter-pixel correlation from previous processing.

The final threshold t_{sg} needs further adjustment to account for the possibility that the signal in the smoothed data has become corrupted, such that the value v_{sg} at $\mathbf{x}$ is no longer from the same distribution as the noisy grey-scale value v_g at $\mathbf{x}$. This can be checked by considering whether these values are further apart than would be expected given the noise levels in the noisy and smoothed grey-scale image:

$$\begin{aligned} r_v &= |v_g - v_{\text{sg}}| \\ r_t &= \frac{1}{2}\left(t_g + t'_{\text{sg}}\right) \\ t_{\text{sg}} &= \begin{cases} \frac{t'_{\text{sg}}}{\text{erf}(0.025l)} & r_v < r_t \\ \frac{t'_{\text{sg}}}{\text{erf}(0.025l)} + 2\left(r_v - r_t\right) & \text{otherwise} \end{cases} \end{aligned} \tag{18}$$

where r_t is the expected range of the difference in noisy and smooth grey-scale value r_v. Increasing t_{sg} if the difference is too high makes the mask element selection depend more strongly on the noisy rather than the smoothed data.

One further consideration for t_{sg} is that, for images with low noise and hence requiring smaller filters (lower filter length l), there is less advantage to using the smoothed grey-scale data in determining the mask elements, rather than the noisy data. Hence, for lower l, t_{sg} is increased according to an ad-hoc $\text{erf}\,()$ function, such that the smoothed data only starts to contribute to the mask design for $l > 5$.

Having generated values of t_{rgb}, t_g and t_{sg} at $\mathbf{x}$ from the global threshold t, elements are included in $w(\mathbf{x})$ where $I(\mathbf{x} + \mathbf{y})$ is within the appropriate threshold of $I(\mathbf{x})$, for all relevant $\mathbf{y} \in w_l$ (a rectangular window of size $l \times l$), and all image domains of $I(\mathbf{x})$. Figure 2 is an example with fairly high noise, and Fig. 3 with lower noise. In the former example, the mask elements are determined almost entirely by the smoothed grey-scale data; in the latter, the noisy grey-scale and RGB data are more significant.

The mask w is different at each location $\mathbf{x}$, but the same set of masks $w(\mathbf{x})$ are used for all colour channels to prevent colour separation in the final image.

2.4 Sensor noise and RAW data

Noise reduction algorithms have been traditionally tested on (notionally) noise-free images with additive white Gaussian noise (AWGN). AWGN is easy to model and guarantees that the ground truth data is known, but is a poor substitute for real image noise. This is particularly the case for images taken using mobile phones, in which sensor noise is often dominated by light levels, leading to a Poisson distribution ('shot'

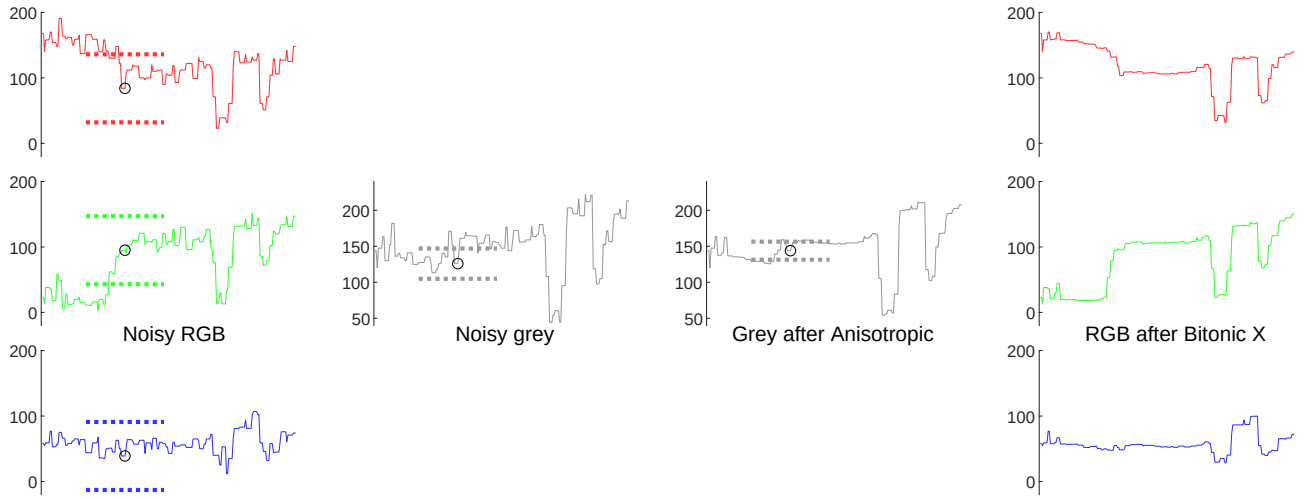


Figure 3: Use of thresholds in Bitonic X to define the locally-adaptive mask $w(\mathbf{x})$. This case has lower noise than Fig. 2, and the noisy RGB and grey-scale images are more significant than the smoothed grey-scale image in determining the mask shape.

noise) with only a small additive Gaussian component ('read' noise) (Tsin et al., 2001). The sensors are overlayed with a colour filter array, where each 2×2 group contains one red, one blue and two green filters in a Bayer pattern which repeats across the entire sensor array. This makes the colour resolution lower than the actual sensor spacing and also leads to varying luminance sensitivity over pixels. RGB colour data, at the original sensor resolution, is created by de-Bayering, or *demosaicing* this data, i.e. interpolating the distributed colour data (Li et al., 2008; Losson et al., 2010). Many demosaicing methods are available, but they all affect the RGB image noise since they introduce correlations between neighbouring RGB pixels.

Standard RGB (sRGB) data is processed in various additional ways, including white-balancing, tone-mapping, and non-linear gamma correction, so that integer intensity values from 0 to 255 can nevertheless efficiently express the logarithmic range to which our eyes are sensitive (Burger, 2010). This strongly accentuates noise at low intensity levels such that the resulting images tend to have higher noise variance at mid to low intensities, even though the reverse is true for the original sensor (RAW) data.

If RAW data is available, it can be filtered directly by considering each of the four elements B_r, B_{g1}, B_{g2} and B_b in the Bayer pattern to form four separate colour channel images, each with one-quarter the original pixels. The 'grey-scale' equivalent B_{grey}, required for local mask definition, is the average of these four channels: the existence of two green channels in this average compensates for their greater importance in measuring luminosity. In this case, the $\sqrt{3}$ in Eq. (15) is also replaced with 2. Since these four channels between them have three times fewer pixels than the demosaiced RGB image, RAW processing is expected to be three times faster than sRGB on the same image with the same filter size l (Park et al., 2009).

For sRGB data, processing is as with AWGN RGB data, except that the application of a non-linear gamma function means that the transformed sensor noise can no longer be considered as zero-mean (Brown and Kim, 2019). This has no effect on the various ranking operations, since gamma correction is order-preserving, but it does bias the smoothing filter $G_{\sigma,\alpha}$. Ranking operations are more efficient with the integer sRGB data, so opening and closing $O_{w,c}$ and $C_{w,c}$ are performed as usual, and the results transformed using a reverse gamma correction just before application of Eqs. (5) to (7). The output $b_{w,c}$ is re-transformed by subsequent use of the forward gamma correction: in both cases, the standard sRGB gamma function is presumed, whether or not precisely this function was applied to the original data.

2.5 Selection of initial threshold

For AWGN noise with known variance σ_n^2, the initial threshold t should be set to $4\sigma_n$, except at low noise levels above a signal-to-noise level of ≈ 22 dB. In this case a smaller global value of t no lower than $2.5\sigma_n$ preserves the signal better, and f' is set to 1 in Eq. (13) since t has already been reduced.

If the variance is not known, there are a variety of ways of estimating this from the initial image. Such methods seek to find parts of the image which contain pure noise, by looking for very low inter-pixel correlation, for instance using Principle Component Analysis (PCA) (Pyatykh et al., 2012). Non-AWGN noise in images can be modelled as having a variance which varies with intensity, in which case similar techniques are employed to estimate the Noise Level Function (NLF) rather than a single variance (Liu et al., 2006; Yang et al., 2015). However, demosaicing of mobile phone data introduces correlations between the sRGB pixels which lead to PCA (or similar approaches) seriously underestimating the amount of noise in an image. A simple way to correct this defect is to measure noise from a sub-image which is sampled every two or three pixels in each direction (Zhou et al., 2020). The noise in such sub-sampled pixels is no longer correlated by demosaicing. Whilst this will also cause a loss in signal correlation, it should be fairly low for typical high resolution images.

Images of σ_n for mobile phone data are hence estimated by first creating $2\times$ sub-sampled images, then taking the abso-

lute value of the discrete second derivative in each direction to remove most of the signal. This is smoothed using a weak $G_{\sigma,\alpha}$ with $\sigma = 1$ and $\alpha = 0.7$, followed by a fixed-shape $R_{w,c}$ with w a circle of diameter 19 and $c = 6\%$, where the low centile tends to select for noise rather than residual signal. The maximum value over each 2×2 Bayer block is kept to account for differences in variance over each colour channel, then scaled to convert to σ_n values.

A variance stabilising transform (Makitalo and Foi, 2012) could be used after inverse gamma correction, to encourage constant variance over intensity. However, the adjusted form of this transform (required to preserve zero-mean noise) is very sensitive to the estimation of precise noise parameters from the image, and any bias introduced to $G_{\sigma,\alpha}$ as a result of poor estimation was found to defeat the slight performance gains due to the variance adjustment. However, to account for the non-constant variance, a slightly higher threshold of $5.5\sigma_n$ was used for these images, where σ_n^2 was the average variance seen in the data.

2.6 Efficient ranking implementation

Ranking operations with fixed masks can be implemented extremely efficiently in nearly constant time with respect to the mask width l (Perreault and Hébert, 2007) with either direct sorting or histogram-based approaches. However, if the mask $w(\mathbf{x})$ is different at each location, the forward ranking operation of Eq. (1) needs to be repeated from scratch at each $\mathbf{x}$, or a sorted list of the superset of possible mask elements covering $l \times l$ can be updated efficiently, and the local masks derived from this list. The former method is better for masks which are expected to have little overlap between neighbouring locations: this is generally the case for low noise scenarios. The latter is quicker where there *is* expected to be a large overlap in mask elements, as is the case in high noise.

The inverse ranking operation of Eq. (2) is updated in the same pass as the forward ranking, by distributing the result of Eq. (1) at $\mathbf{x}$ to all pixels $\mathbf{x} + \mathbf{y}$ for $\mathbf{y} \in w(\mathbf{x})$. A sorted list is updated at each $\mathbf{x} + \mathbf{y}$ with these new values: once the forward operation has finished, the reverse ranking simply involves selecting the appropriate centiles from these sorted lists. Equations (3) and (4) only require fairly low or high centiles: so only the tails (below c and above $100-c$) of these sorted lists are updated, saving on both time and memory. Nevertheless, this approach would require storage of $\approx 2 \times 0.08 \times l^2 \times |I|$ for each forward ranking operation, which is still considerable for large image size $|I|$, and mask size l. A significant reduction can further be achieved by generating the reverse ranking result at $\mathbf{z}$ as soon as the forward mask $\mathbf{x}$ has moved beyond the point where any further update to the sorted list at $\mathbf{z}$ is possible.

Since the same mask $w(\mathbf{x})$ is used for all colour channels, it is more efficient to perform the ranking operations for all channels in parallel. In that case, $w(\mathbf{x})$ can be calculated once at each $\mathbf{x}$, used for forward and inverse ranking on all channels, and immediately discarded.

3 Results

3.1 Images with AWGN

The fixed-shape bitonic was compared to various linear and morphological filters in (Treece, 2016), with clear improvement relative to all other morphological-based filters, e.g. the OCCO filter (Aptoula and Lefevre, 2007), and self-dual filters (Caselles and Monasse, 2002; Monasse and Guichard, 2000; Maragos and Evangelopoulos, 2007; Serra et al., 2013). The structurally-varying Bitonic V was demonstrated in (Treece, 2019) to outperform Non-Local Means (Buades et al., 2005), anisotropic diffusion (Perona and Malik, 1990) and image-guided or bilateral filters (He et al., 2013; Tomasi and Manduchi, 1998), but not Block-matching (Dabov et al., 2007), which is generally accepted to be the reference standard for non-learning-based filters.

The Bitonic X filter is here assessed on the same convenience sample of 23 images as in (Treece, 2019), but with increased signal-to-noise ratio (SNR) range from ≈36 dB (very low noise) to −6 dB (very high noise). These include standard test images from public-domain sites[1], various high dynamic range (HDR) images all with the CC0 Creative Commons licence, and two simple computer-generated images.

The full list of non-learning-based filters tested, where σ_n^2 is the added noise variance, is:

BM3D Block-matching[2] (Dabov et al., 2007), with σ set to a variety of trial values $\sigma = \frac{10+l}{20}\sigma_n$, and the default profile, i.e. 'normal' ($\sigma < 0.16$), or 'vn' ($\sigma \geq 0.16$).

NLM Non-local means filter, using a fast MATLAB implementation[3] (Buades et al., 2005), with varying l for window and search length, and the parameter h set to σ_n.

Diffusion Anisotropic diffusion (Perona and Malik, 1990), using a fast MATLAB implementation[4], with iterations $n = \frac{l}{2}$, integration constant set to σ_n, gradient threshold set to $2\sigma_n$, and the wide-region conduction coefficient.

Anisotropic The anisotropic Gaussian filter $G_{\sigma,\alpha}$, with varying size l, $\sigma = 0.21l$ and $\alpha = 0.6$.

Bitonic Fixed bitonic filter as in (Treece, 2016), with varying mask diameter l, $c = 10\%$ and t set to $2.8\sigma_n$.

Bitonic V Structurally varying bitonic filter, as in (Treece, 2019), with varying mask width l, $c_1 = 4\%$, $\alpha = 0.6$ and t set to $2.8\sigma_n$.

Bitonic MV Multi-resolution version of the above, with three levels.

Bitonic X Locally-adaptive bitonic filter described in this paper, with varying mask width l, $c = 8\%$, $\alpha = 0.6$ and t set to $4\sigma_n$ (except for the lowest noise levels, where t approached $2.5\sigma_n$).

[1] including http://decsai.ugr.es/cvg/CG/base.htm

[2] MATLAB BM3D v2.0 software from http://www.cs.tut.fi/~foi/GCF-BM3D/

[3] Dirk-Jan Kroon (2021). Fast Non-Local Means 1D, 2D Color and 3D (https://www.mathworks.com/matlabcentral/fileexchange/27395-fast-non-local-means-1d-2d-color-and-3d), MATLAB Central File Exchange. September 7, 2010.

[4] Daniel Lopes (2021). Anisotropic Diffusion (Perona & Malik) (https://www.mathworks.com/matlabcentral/fileexchange/14995-anisotropic-diffusion-perona-malik), MATLAB Central File Exchange. May 16, 2007.

Bitonic MX Multi-resolution version of the above, with five levels.

Median X A median filter with the same spatially-varying masks $w(\mathbf{x})$ but just using $R_{w,c}$ with $c = 50\%$.

In each case, the parameter l was optimised over individual images and noise levels for the best joint SNR and SSIM performance, with the other parameters fixed over all images.

Results are summarised for all images over the entire noise range in Fig. 4, with examples in Figs. 5 and 6, where σ_n is expressed relative to the maximum range of the original image data.

3.2 Images with real noise

The Smartphone Image Denoising Dataset (SIDD) (Abdelhamed et al., 2018) and the Darmstadt Noise Dataset (DND) (Plotz and Roth, 2017) were used to assess noise reduction on more realistic images. The SIDD data is of slightly higher quality and also involves higher noise levels than DND, but the latter includes a greater range of images for both smartphones and DLSR cameras. The more recent Natural Image Noise Dataset (NIND) (Brummer and De Vleeschouwer, 2019) contains DLSR-like images only, as does the slightly older Renoir dataset (Anaya and Barbu, 2018).

The SIDD data consists of 10 different scenes captured with five different smartphones each under four combinations of 15 different ISO levels, three illumination temperatures and three brightness levels. The validation and benchmark data sets both consisted of 40 images from this set of 200, each with 32 (different) randomly selected non-overlapping image patches of size 256×256 pixels. The validation data contained ground-truth images for these patches, whereas no ground truth data was released for the benchmark patches: these could only be assessed by submission of results to the website[5]. Initial results, consisting of Peak SNR (PSNR), SSIM and processing time averaged over all benchmark patches, were provided in (Abdelhamed et al., 2018), with further results from a recent competition in (Abdelhamed et al., 2019).

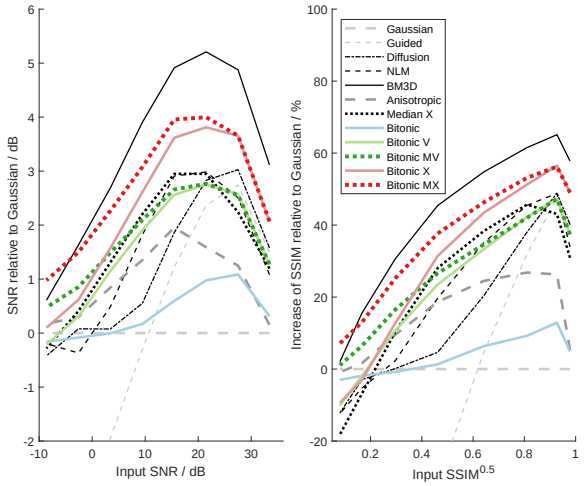
The DND data consists of 50 different scenes captured with four different camera lenses and a range of ISO levels. The benchmark data set consisted of 20 image patches of size 512×512 pixels from each of these 50 images, with at most 10% overlap. No ground truth data was available: assessment was only by submission of results to the website[6]. Initial results, consisting of PSNR, SSIM and processing time per patch, were provided in (Plotz and Roth, 2017), with many additional results, together with relevant citations, on the associated website.

Additional filters tested on these images follow the discussion in Section 2.4:

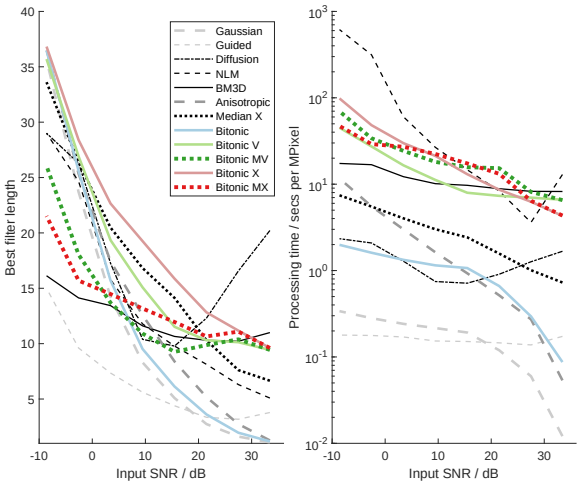
— **G** A version of the filter which includes an inverse-gamma correction before filtering, followed by re-correcting afterwards.

— **Raw** The filter applied to RAW instead of sRGB data.

[5] https://www.eecs.yorku.ca/~kamel/sidd/
[6] https://noise.visinf.tu-darmstadt.de/



(a) SNR and SSIM performance



(b) Filter length and processing time

Figure 4: Summarised results over the complete set of AWGN images. Performance (after optimisation of filter extent l in each case) is averaged over each noise level in each image. (a) SNR and SSIM results shown relative to the best performance of a Gaussian filter. (b) Average filter parameter l and processing times for these performance results.

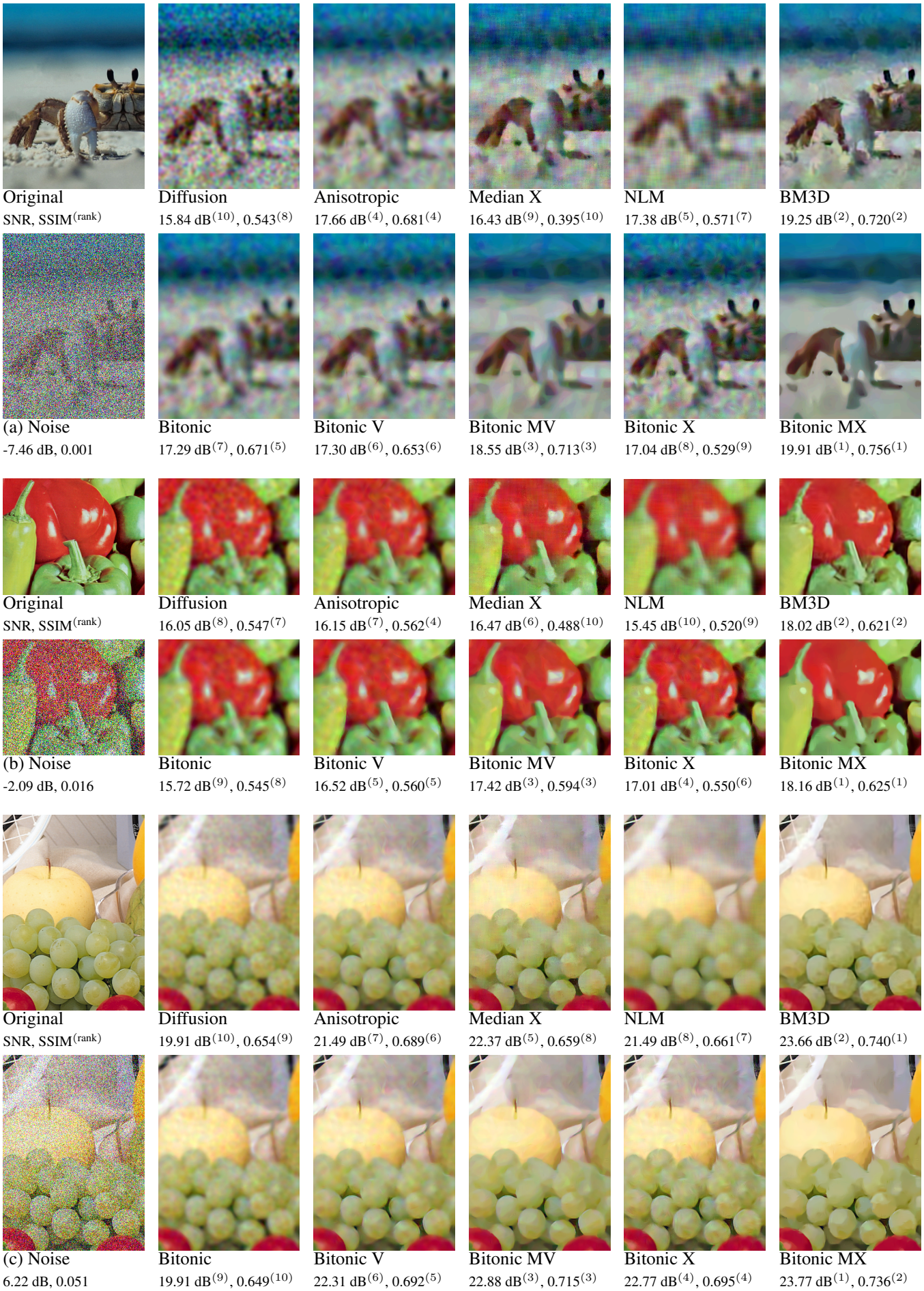


Figure 5: Results on AWGN images (a) 'south sound' with $\sigma_n = 1.28$ added noise, (b) 'peppers' with $\sigma_n = 0.64$ added noise and (c) 'fruits' with $\sigma_n = 0.32$ added noise.

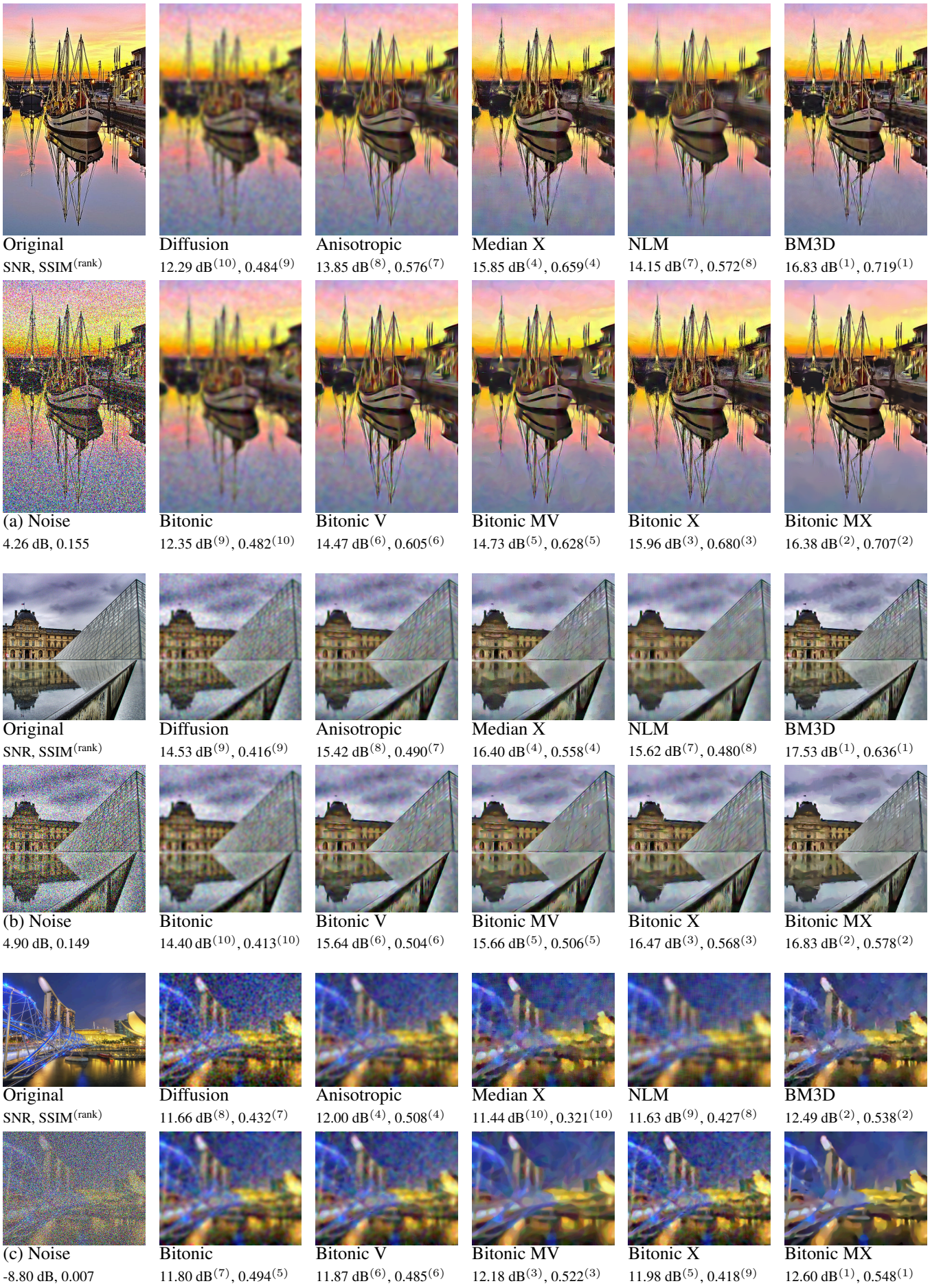


Figure 6: Results on AWGN images (a) 'del presepe' with $\sigma_n = 0.32$ added noise, (b) 'louvre' with $\sigma_n = 0.32$ added noise and (c) 'marina bay' with $\sigma_n = 1.28$ added noise.

In this case, the Bitonic X filters also all include an inverse-gamma correction after the rank filters when applied to sRGB data. The SIDD validation data set was used to compare all the relevant filters, since the patches could be assessed independently and used to display results against SNR; the benchmark data was also used for the key Bitonic MX and updated BM3D filters. For SIDD data, the noise variance σ_n^2 was estimated from each image as in Section 2.5, and the single parameter l was optimised for the validation data set, but then used un-changed on the benchmark data set. For DND data, σ_n was provided, and l was fixed over all images, to $l = 15$ for sRGB and $l = 13$ for RAW data, following the results seen in Fig. 7(b) for SIDD.

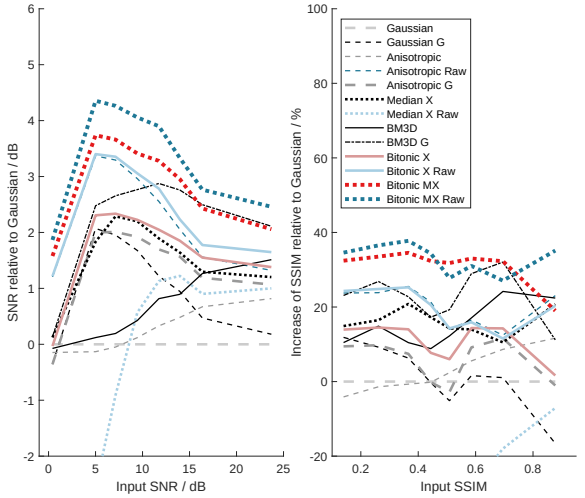
Validation and benchmark results are summarised over all images in each data set in Table 1, with validation results from the SIDD data displayed over SNR in Fig. 7. Some examples of specific image patches for the SIDD data are in Figs. 8 and 9 and for the DND data in Fig. 10.

4 Discussion

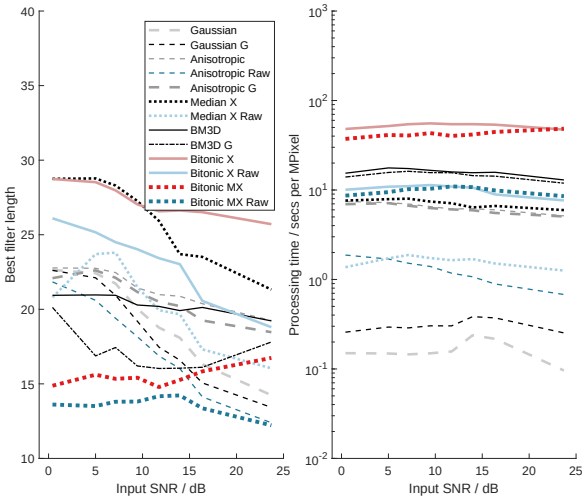
The AWGN results for non-learning based filters, which are a continuation of those from the previous paper (Treece, 2019), are considered first. It is clear from Fig. 4 that the Bitonic MX is an improvement over the previous bitonic filters on all counts, with better SNR and SSIM at all noise levels, and reduced processing time on average. BM3D still performs very well on this type of noise, but as expected the Bitonic MX is better for high noise scenarios. In these cases, smoothly varying parts of images (particularly noticeable in the background, for instance of Figs. 5(a) and 6(a)) are much better preserved than in BM3D: a direct result of using only local image data rather than searching further afield for patterns.

It is also clear that the various components of the new filter all contribute significantly to the overall performance. Whilst there are some occasions where the Anisotropic, Median X or Bitonic X perform well on their own, the summary results are considerably worse than for Bitonic MX. It is, however, notable that the Median X in particular is much faster and may well be of use for some situations, with a performance which is surprisingly good for a median-based filter.

The results on real image noise are intriguing in that, relative to BM3D, the Bitonic MX is better than would have been expected given the performance across AWGN noise. Whilst the real image noise in both the SIDD and DND data sets is substantially lower than the extreme levels of AWGN noise modelled, the results seem to follow those for relatively higher levels of AWGN noise. This is particularly apparent when comparing the summary graphs in Fig. 4 and Fig. 7, with the new filter outperforming BM3D at all but the lowest noise levels, even given the modifications which considerably improve on the previously reported BM3D results. There are several possible reasons for this. Firstly, real images tend to have quite high sampling resolution (number of pixels) compared to the optical resolution (sharpness of focus) and actual signal frequencies (amount of fine detail). Hence there are potentially larger groups of pixels dominated by noise, which the Bitonic MX is particularly good at handling. Secondly, the SNR as measured in the sRGB image is likely to be somewhat lower than the real SNR, since the camera processing pipeline acts to reduce the level of the noise, but increase



(a) SNR and SSIM performance



(b) Filter length and processing time

Figure 7: Summarised results over the SIDD validation images. Performance (after optimisation of filter extent l in each case) is averaged and grouped according to the noise level in each image patch. (a) SNR and SSIM results shown relative to the best performance of a Gaussian filter. (b) Average filter parameter l and processing times for these performance results.

Evaluation space	RAW			sRGB				
	SNR dB	**PSNR** dB	**SSIM**	**SNR** dB	**PSNR** dB	**SSIM**	**Time** secs	**Param**
Results on SIDD validation data set								
Noise	9.32	37.19	0.837	10.82	23.65	0.484	-	-
Gaussian Raw	20.77	48.64	0.983	22.48	35.31	0.912	0.06	13.4
Gaussian	-	-	-	21.12	33.94	0.907	0.16	19.2
Gaussian G	-	-	-	22.14	34.96	0.912	0.31	18.4
Anisotropic Raw	21.74	49.61	0.987	23.41	36.23	0.923	1.29	17.4
Anisotropic	-	-	-	21.38	34.21	0.908	6.45	21.4
Anisotropic G	-	-	-	22.50	35.33	0.915	6.20	20.8
Median X Raw	19.82	47.69	0.973	20.60	33.43	0.854	1.60	20.4
Median X	-	-	-	22.68	35.50	0.918	7.13	25.5
Bitonic X Raw	21.79	49.66	0.987	**23.55**[3]	**36.38**[3]	0.925	10.18	23.2
Bitonic X	-	-	-	22.82	35.65	0.919	52.44	27.2
Bitonic MX Raw	22.41	50.28	0.989	**24.49**[1]	**37.32**[1]	**0.936**[2]	9.85	13.6
Bitonic MX	-	-	-	**24.00**[2]	**36.83**[2]	**0.936**[1]	42.15	15.4
BM3D	-	-	-	21.76	34.59	0.923	15.90	19.4
BM3D G	-	-	-	23.40	36.23	**0.931**[3]	14.71	16.0
Results on SIDD benchmark data set								
Noise	-	36.75	0.840	-	23.68	0.480	-	-
Bitonic MX Raw	-	49.41	0.987	-	**37.25**[2]	**0.934**[2]	8.91	13.6
Bitonic MX	-	-	-	-	**36.67**[3]	**0.933**[3]	42.2	15.4
BM3D G	-	-	-	-	36.20	0.929	17.3	16.0
BM3D (initial)	-	45.52	0.98	-	30.95	0.863	34.3	-
NLM (initial)	-	44.06	0.971	-	29.39	0.846	210.7	-
UPI (for DND)	-	49.17	0.974	-	-	-	-	-
CycleISP (for DND)	-	50.14	0.976	-	-	-	-	-
CycleISP	-	52.41	0.993	-	**39.56**[1]	**0.956**[1]	-	-
Results on DND benchmark data set								
Noise	-	39.40	0.863	-	29.83	0.701	-	-
Bitonic MX Raw	-	46.61	0.971	-	**38.02**[3]	**0.936**[3]	7.6	13.0
Bitonic MX Raw (b)	-	46.57	0.969	-	37.99	0.935	7.1	13.0
Bitonic MX	-	-	-	-	37.85	**0.936**[3]	30.0	15.0
BM3D G	-	-	-	-	37.87	0.934	13.9	16.0
BM3D	-	-	-	-	36.79	0.912	13.7	20.0
BM3D Raw (initial)	-	46.64	0.972	-	37.78	0.931	-	-
BM3D (initial)	-	-	-	-	34.51	0.851	-	-
UPI	-	48.89	0.982	-	**40.35**[2]	**0.964**[2]	-	-
CycleISP	-	49.13	0.983	-	**40.50**[1]	**0.966**[1]	-	-

Table 1: Summarised results for SIDD and DND images. The time is CPU time per mega-pixel image and **Param** is the average filter extent l. Results in *italics* are from previous papers (Brooks et al., 2019; Zamir et al., 2020) and the SIDD and DND websites. The (for DND) results were trained on DND but applied to SIDD and hence show more realistic performance on unseen data. Bitonic MX Raw (b) used noise variance from the image data instead of the provided value. CycleISP represents the best performing deep-learning approach on RAW and sRGB data across the SIDD and DND images.

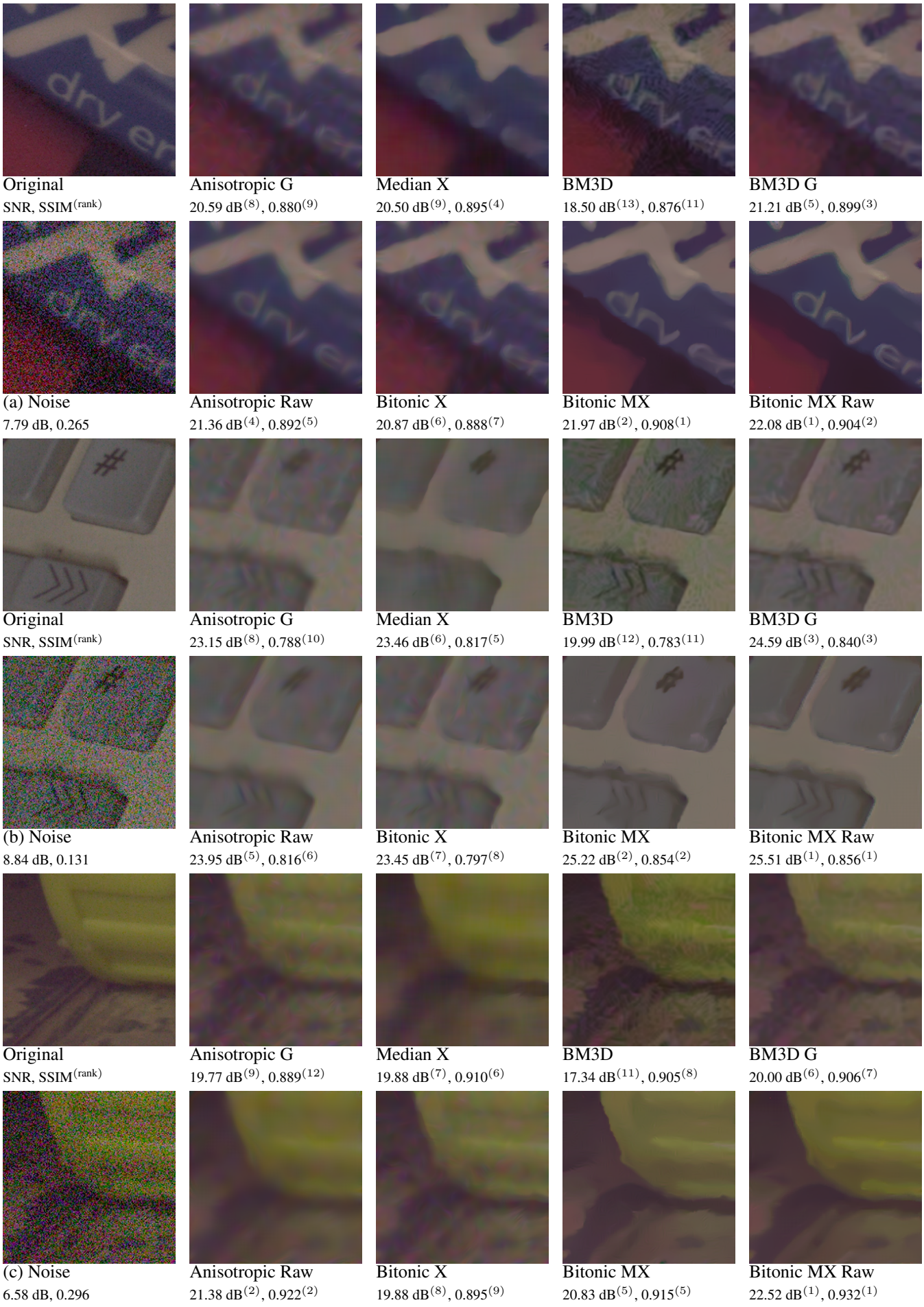


Figure 8: Results on sample SIDD images (blocks) (a) 6 (5), (b) 11 (11) and (c) 28 (14).

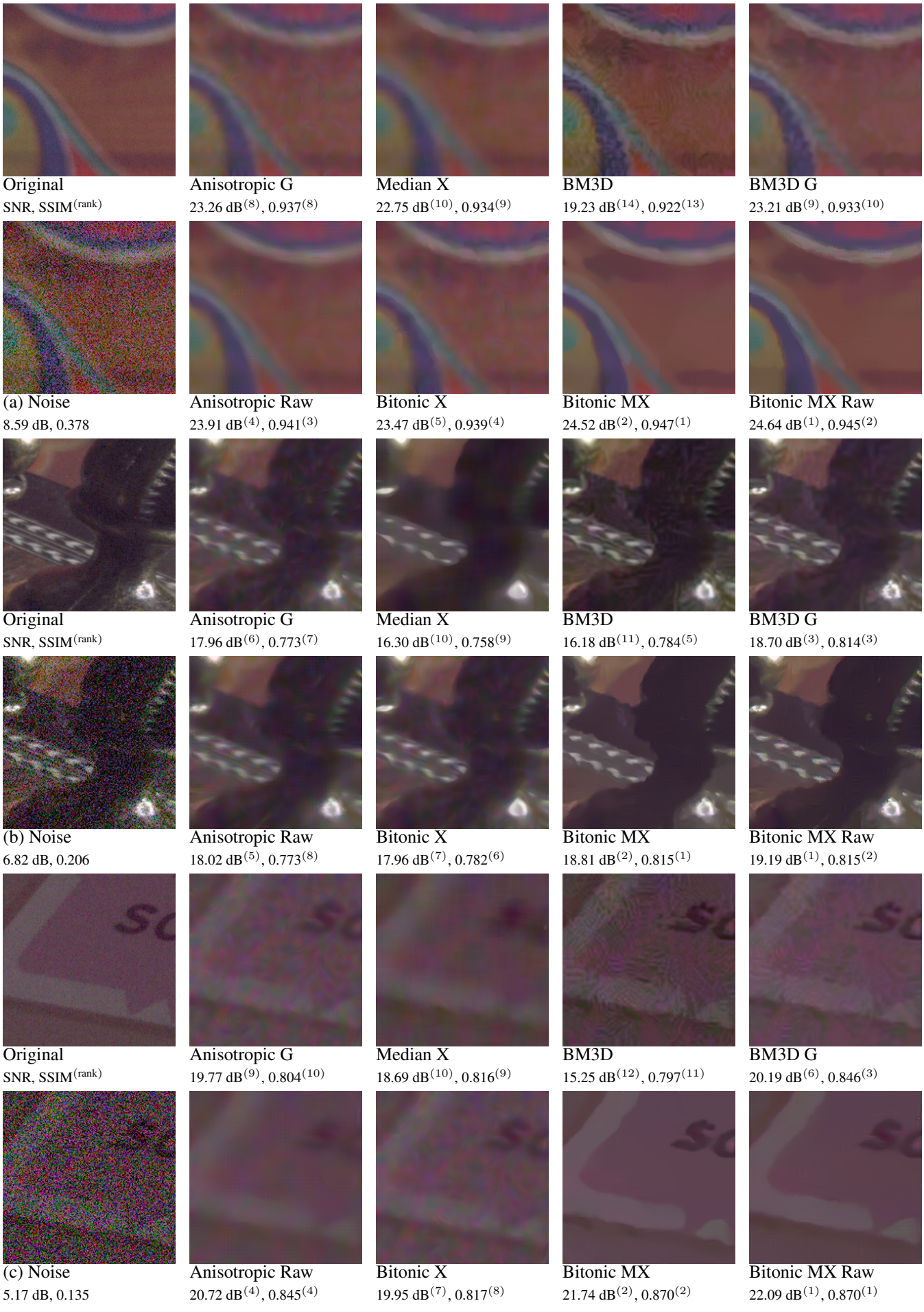


Figure 9: Results on sample SIDD images (blocks) (a) 19 (2), (b) 19 (29) and (c) 33 (13).

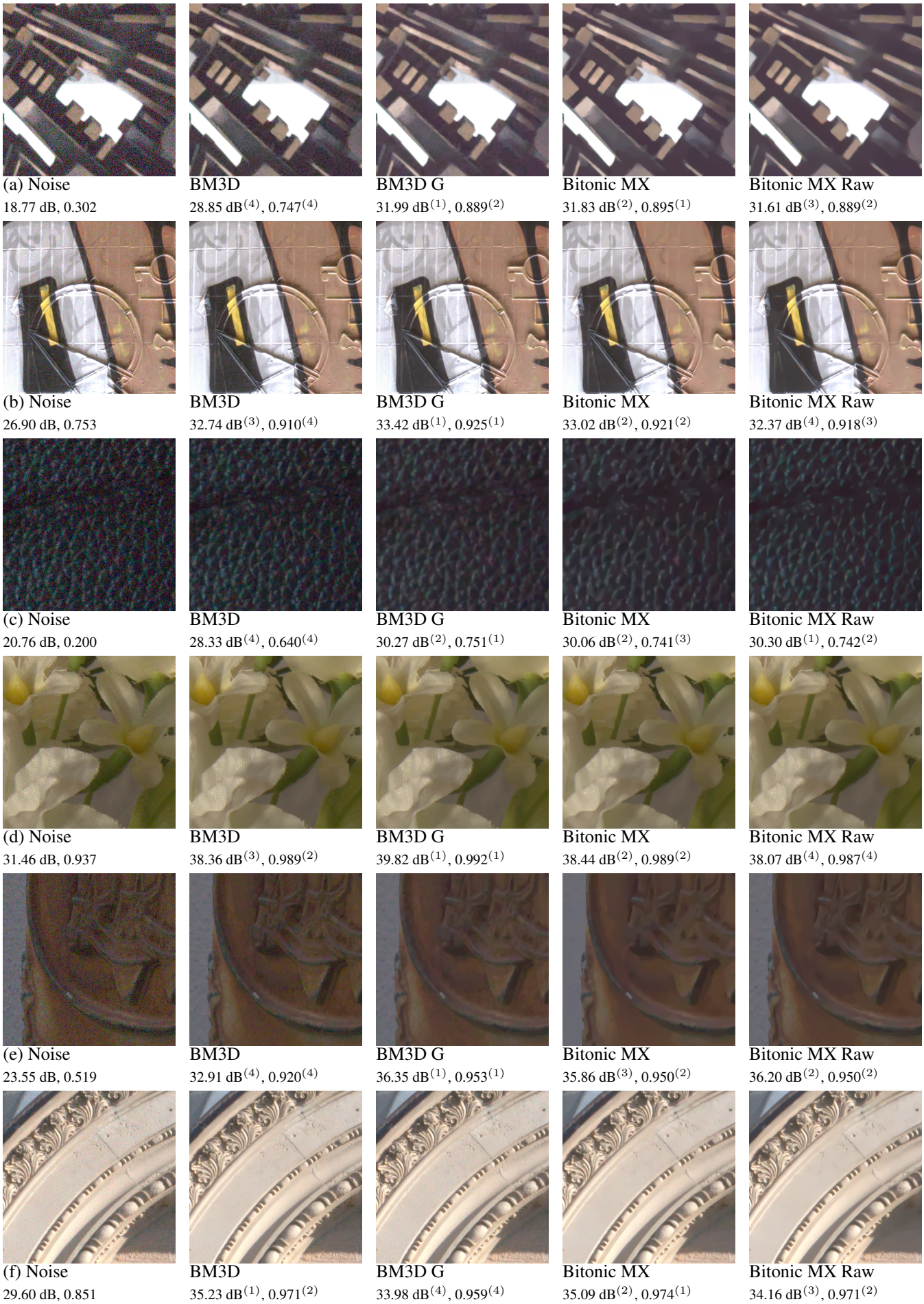


Figure 10: Results on sample DND images (blocks) (a) 1 (18), (b) 2 (19), (c) 16 (9), (d) 17 (3), (e) 26 (2) and (f) 44 (1).

the noise correlation between pixels. In effect, the real noise level is actually higher than it appears. Thirdly, the novel use of bitonicity to differentiate noise from signal may allow the new filter to apply more generally to the more complex, correlated, real image noise than does BM3D.

The SIDD validation set again shows that all the parts of the Bitonic MX are necessary for good performance. Where RAW data is available, the results are generally better, with considerably reduced processing times, but the performance difference between RAW and sRGB is not huge, with both better than BM3D on this and the benchmark data. Surprisingly, this is also the case for the DND benchmark, even though this has noise levels which are about 6 dB lower on average. Here the improvement between BM3D and Bitonic MX on sRGB data is ambiguous, with better SSIM but slightly worse PSNR, but the sRGB results when the filter is applied to RAW data are consistently higher.

Whilst the new filter is a clear improvement over other non-learning-based filters on the SIDD and DND data sets, it is equally clear that learning-based approaches can perform better. CycleISP (Zamir et al., 2020) is a good example of a very strong performer on both data sets, but there are many other alternatives which come close to these results (Abdelhamed et al., 2019). *How much* better these techniques can perform on this data is a fair question, however. Until now, BM3D performance on sRGB images was reported as 30.95 dB on SIDD and 34.51 dB on DND. With the use of a simple and fast inverse gamma transform, these results increase to 36.20 dB and 37.87 dB, respectively, both much higher than has previously been reported. On the DND data set, where the noise variance is known, this makes the result similar to that for RAW processing (37.78 dB). Previous under-performance on the SIDD data set is probably also due to inappropriate setting of σ_n from PCA analysis (or similar) of sRGB data which is heavily compromised due to the noise correlation on this data. The best Bitonic MX performance is 37.25 dB and 38.02 dB, which is on average 2.4 dB lower than the best learning-based approaches: a significant gap, but somewhat smaller than has previously been reported, and better than all techniques reported in the original papers for these datasets (Abdelhamed et al., 2018; Plotz and Roth, 2017) as well as many learning-based results on the associated websites.

On the generality of learning-based techniques, it is interesting to note the results from the CycleISP paper (Zamir et al., 2020), which apply their method and another based on un-processing (UPI) (Brown and Kim, 2019), both trained on just the DND data, directly to the SIDD data. As seen in Table 1, this immediately reduces the performance to a similar level to that for Bitonic MX: in fact Bitonic MX achieves better SSIM values than either of these tests, and an intermediate PSNR value. Other authors have noted the particular training issues with the SIDD data (Kim et al., 2020). Whilst the actual benchmark ground truth images are not available, the training data set includes exactly the same images captured with the same cameras, but in different conditions, and other images captured with the same cameras in the same conditions. In addition, the validation data contains different patches to the benchmark data, but from precisely the same images as the benchmark data. Much of the performance of learning-based techniques is derived from very specific training which may not always be possible.

In contrast, Bitonic MX has very few parameters. The few constants mentioned in this paper are genuinely constant over all the tested data sets. Such constants exist in most algorithms, including BM3D. That leaves only the threshold t and filter size l as tunable parameters. Whilst Fig. 4 shows that somewhat smaller filter sizes do tend to generate better results for AWGN data with low noise, it can be seen from Fig. 7 that a filter length $l = 15$ is generally a good value for sRGB data, and indeed fixing l across all images for the DND data still gave good results. The threshold t remains an important parameter, in much the same way as the setting of σ is critical for BM3D. The SIDD results show that it is possible to estimate this sensibly, with no prior knowledge, from the image data; the DND and AWGN results show it can alternatively be set given knowledge of the overall noise variance. Prior knowledge of noise variance is not, however, critical: using the estimated noise variance rather than the given value for the RAW DND data (Bitonic MX Raw (b) in Table 1) resulted in almost identical performance.

5 Conclusion

The new Bitonic MX filter is a significant improvement over the previous version, performing better than BM3D for high levels of AWGN noise and, more importantly, better across a broad range of noise levels for real images from both the SIDD and DND data sets. This is still the case even allowing for a much improved performance of BM3D following the addition of an inverse gamma transformation. Bitonic MX is the best performing non-learning technique on these data sets, narrowing the gap between this and the best learning technique to 2.4 dB on average. The results are now similar to that achieved from learning-based technique when not directly trained on data from the same sources. They demonstrate what can be achieved with a predictable, explainable, entirely local technique, which makes no assumptions of repeating patterns either within an image or across images, and hence creates residual images which are well behaved even in very high noise.

Implementations of all the novel filters in this paper are available for Matlab[7] and also for Windows in wxDicom[8] software.

Acknowledgements

Thanks to my Masters student Léa Gansser-Potts for demonstrating that there was some potential in developing threshold-based masks.

References

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

[7] Graham Treece (2021). Locally Adaptive Bitonic X Filter (https://www.mathworks.com/matlabcentral/fileexchange/98899-locally-adaptive-bitonic-x-filter), MATLAB Central File Exchange. September 8, 2021.

[8] http://mi.eng.cam.ac.uk/Main/GMT_wxDicom

Abdelhamed, A., Timofte, R., Brown, M. S., 2019. NTIRE 2019 challenge on real image denoising: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Anaya, J., Barbu, A., 2018. Renoir–a dataset for real low-light image noise reduction. Journal of Visual Communication and Image Representation 51, 144–154.

Aptoula, E., Lefevre, S., 2007. A comparative study on multivariate mathematical morphology. Pattern Recognition 40 (11), 2914–2929.

Briggs, W. L., Henson, V. E., McCormick, S. F., 2000. A multigrid tutorial, 2nd Edition. SIAM.

Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., Barron, J. T., 2019. Unprocessing images for learned raw denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11036–11045.

Brown, M. S., Kim, S., 2019. Understanding the in-camera image processing pipeline for computer vision. In: IEEE International Conference on Computer Vision (ICCV)-Tutorial. Vol. 3.

Brummer, B., De Vleeschouwer, C., 2019. Natural image noise dataset. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.

Buades, A., Coll, B., Morel, J.-M., 2005. A review of image denoising algorithms, with a new one. Multiscale Modeling & Simulation 4 (2), 490–530.

Burger, W., 2010. Color space considerations for linear image filtering. In: Proc. 34th Workshop of the Austrian Ass. for Pattern Recognition, Zwettl, Austria. Citeseer, pp. 163–170.

Caselles, V., Monasse, P., 2002. Grain filters. Journal of Mathematical Imaging and Vision 17 (3), 249–270.

Catmull, E., Rom, R., 1974. A class of local interpolating splines. In: Barnhill, R., Risenfeld, R. (Eds.), Computer Aided Geometric Design. Academic Press, San Francisco, pp. 317–326.

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

Gu, S., Timofte, R., 2019. A brief review of image denoising algorithms and beyond. Inpainting and Denoising Challenges, 1–21.

He, K., Sun, J., Tang, X., 2013. Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (6), 1397–1409.

Kass, M., Solomon, J., 2010. Smoothed local histogram filters. ACM Transactions on Graphics 29 (4), Article 100.

Kim, Y., Soh, J. W., Park, G. Y., Cho, N. I., 2020. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3482–3492.

Li, X., Gunturk, B., Zhang, L., 2008. Image demosaicing: A systematic survey. In: Visual Communications and Image Processing 2008. Vol. 6822. International Society for Optics and Photonics, p. 68221J.

Liu, C., Freeman, W. T., Szeliski, R., Kang, S. B., 2006. Noise estimation from a single image. In: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06). Vol. 1. IEEE, pp. 901–908.

Losson, O., Macaire, L., Yang, Y., 2010. Comparison of color demosaicing methods. Advances in Imaging and Electron Physics 162, 173–265.

Makitalo, M., Foi, A., 2012. Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. IEEE transactions on image processing 22 (1), 91–103.

Maragos, P., Evangelopoulos, G., 2007. Leveling cartoons, texture energy markers, and image decomposition. In: Proceedings of the Eighth International Symposium on Mathematical Morphology. ISMM '07. Kluwer Academic Publishers, pp. 125–138.

Monasse, P., Guichard, F., 2000. Fast computation of a contrast-invariant image representation. IEEE Transactions on Image Processing 9 (5), 860–872.

Park, S. H., Kim, H. S., Lansel, S., Parmar, M., Wandell, B. A., 2009. A case for denoising before demosaicking color filter array data. In: 2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers. IEEE, pp. 860–864.

Perona, P., Malik, J., 1990. Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12 (7), 629–639.

Perreault, S., Hébert, P., 2007. Median filtering in constant time. IEEE Transactions on Image Processing 16 (9), 2389–2394.

Plotz, T., Roth, S., 2017. Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1586–1595.

Pyatykh, S., Hesser, J., Zheng, L., 2012. Image noise level estimation by principal component analysis. IEEE transactions on image processing 22 (2), 687–699.

Serra, J., Vachier, C., Meyer, F., 2013. Levelings. In: Mathematical Morphology: From Theory to Applications. Wiley, pp. 199–228.

Tian, C., Fei, L., Zheng, W., Xu, Y., Zuo, W., Lin, C.-W., 2020. Deep learning on image denoising: An overview. Neural Networks.

Tomasi, C., Manduchi, R., 1998. Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, 1998. IEEE, pp. 839–846.

Treece, G. M., Nov. 2016. The bitonic filter: linear filtering in an edge-preserving morphological framework. IEEE Transactions on Image Processing 25 (11), 5199–5211.

Treece, G. M., Aug. 2019. Morphology-based noise reduction: structural variation and thresholding in the bitonic filter. IEEE Transactions on Image Processing 29, 336–350.

Tsin, Y., Ramesh, V., Kanade, T., 2001. Statistical calibration of ccd imaging process. In: Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001. Vol. 1. IEEE, pp. 480–487.

Van Vliet, L. J., Verbeek, P. W., 1995. Estimators for orientation and anisotropy in digitized images. In: ASCI. Vol. 95. pp. 16–18.

Yang, J., Gan, Z., Wu, Z., Hou, C., 2015. Estimation of signal-dependent noise level function in transform domain via a sparse recovery model. IEEE Transactions on Image Processing 24 (5), 1561–1572.

Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., Yang, M.-H., Shao, L., 2020. Cycleisp: Real image restoration via improved data synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2696–2705.

Zhou, Y., Jiao, J., Huang, H., Wang, Y., Wang, J., Shi, H., Huang, T., 2020. When awgn-based denoiser meets real noises. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. pp. 13074–13081.