Real Image Denoising with a Locally-Adaptive Bitonic Filter

Graham Treece

Abstract—Image noise removal is a common problem with many proposed solutions. The current standard is set by learning-based approaches, however these are not appropriate in all scenarios, perhaps due to lack of training data or the need for predictability in novel circumstances. The bitonic filter is a non-learning-based filter for removing noise from signals, with a mathematical morphology (ranking) framework in which the signal is postulated to be locally bitonic (having only one minimum or maximum) over some domain of finite extent. A novel version of this filter is developed in this paper, with a domain that is locally-adaptive to the signal, and other adjustments to allow application to real image sensor noise. These lead 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. It also surpasses this and other more recent non-learning-based filters for two public data sets containing real image noise at various levels. This is despite an additional adjustment to the block-matching filter, 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 2.4 dB lower than the best learning-based techniques when they are optimally trained. However, 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.

Index Terms—bitonic filter, morphology, noise reduction, sensor noise

I. 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 [1], [2]. 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 each technique. 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 [3], [4], [5].

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 [6] then took a structurally-varying form [7]. 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) [8] at very high noise levels.

In previous work, it was suggested that further improvements may be possible from the use of more complex domains (mask shapes) [7]. The novel design of such domains is the primary focus of this paper. However, previous results were all based on adding simulated noise to noise-free images: typical images from modern smartphones or larger lens cameras exhibit different properties, and have consequently shown quite different performances across noise-reduction techniques [3], [5]. Hence this paper also develops the new filter for real noise and investigates performance on two recently developed public datasets which are based on such images.

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 BM3D and more recent non-learning-based filters on real image noise, it has also become clear that the performance of BM3D has been somewhat under-reported in this scenario, with some simple but novel adjustments considerably improving on previously published results.

II. RELATED WORK

The existing bitonic filter is briefly summarised in Sections II-A and II-B, since it is not well known. The new filter...
developed within this framework is outlined in Section III.

A. Bitonic filter

The constant-shape bitonic filter was presented in detail in [6] and extended to a structurally-varying version in [7]. See these papers for a fuller description of the rationale, but essentially it consists of a robust opening $O_{w,c}$ and closing $C_{w,c}$ of a signal $I(x)$ which varies with location $x$:

$$R_{w,c}(I)(x) = \text{c-th centile \{I(x + y)\}}$$

(1)

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

(2)

$$O_{w,c} = R_{w,100-c}(R_{w,c}(I))$$

(3)

$$C_{w,c} = R_{w,100-c}(R_{w,c}(I))$$

(4)

where $R_{w,c}$ is a rank filter, and $y$ is a vector offset to a location close to $x$, 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. This centile is set to less than 10%: 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 [9] gives more robust results in the presence of noise.

All $|w|$ mask elements are constrained to be within a square region of size $l \times l$. In [6] $w$ was a circle of diameter $l$, constant for all $x$, whereas in [7] $w$ at each $x$ was chosen from a small set of elliptical masks with varying orientations and aspect ratios. In the latter case a global threshold $t$ was used to further exclude mask elements $y$ where $|I(x) - I(x + y)| > t$. For constant mask shapes $w$, the reverse rank filter $R^{-1}_{w,c}$ is the same as $R_{w,c}$, i.e. $w$ at $x$ must contain $y$ for inclusion of data from $x + y$. When evaluating $R_{w,c}$ for varying masks, what matters is whether the masks used in the forward rank at surrounding locations $x + y$ overlap with $x$, i.e. contain $-y$.

Opening and closing operations perform well since they are insensitive to data values, only to the ordering of the data. However, they 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 $G_{\sigma,\alpha}$.

$$\epsilon_O = |G_{\sigma,\alpha} (I(x) - O_{w,c}(x))|$$

(5)

$$\epsilon_C = |G_{\sigma,\alpha} (C_{w,c}(x) - I(x))|$$

(6)

$$b_{w,c} = \frac{\epsilon_O^2 (C_{w,c} - \epsilon_C) + \epsilon_C^2 (O_{w,c} + \epsilon_O)}{\epsilon_O^2 + \epsilon_C^2}$$

(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 $\epsilon_O$ and $\epsilon_C$ 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 $G_{\sigma,\alpha}$ in Eqs. (5) and (6) is sensitive to the local direction $\phi(x)$ and degree of anisotropy $\gamma(x)$ in the image $I(x)$ which, following [10], 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). $\phi(x)$ is the angle following the dominant features in the image, i.e. the direction in which the filter should be aligned. These are used to construct a Gaussian-like filter which has a spatial range given by $\sigma$ (here set to 0.21l), and rotational asymmetry $\alpha$ (set to 0.6), controlling the extent to which the filter follows the dominant image direction and anisotropy:

$$d_y(x) = |y| \sin(\gamma(y) - \phi(x))$$

(8)

$$\phi(x) = \sum_{y \in w_l} \phi_y(x) I(x + y)$$

(9)

$$\phi_{y}(x) = \sum_{y \in w_l} \phi_{y}(x) I(x + y)$$

(10)

In this case $y$ is restricted to a square mask $w_l$ of size $l \times l$. A data threshold can also be used to set $\phi_y(x) = \Omega_y(x)(x+y) = 0$ for pixels with $|I(x) - I(x + y)|$ too large.

For colour images, $\phi(x)$ and $\gamma(x)$ are calculated from the noisy greyscale image. Subsequent operations are performed independently for each colour channel, but application of Eqs. (5) and (6) all use the same $\phi(x)$ and $\gamma(x)$ to ensure there is no colour separation due to smoothing in different directions.

B. Multi-resolution framework

In [7], the bitonic filter was set in a multi-resolution framework. At each level $n$, the processed result from the higher level is reduced to quarter size, and bitonic filtering applied to this processed, reduced, image. The global threshold $t$, which affects the masks $w$ during filtering, is set to $t_n$ at each level: this is lower for deeper levels, since the noise variance will have been reduced due to the previous processing. Figure 1 summarises the approach. Reduction of the image at each level is achieved with a Catmull-Rom spline [11] restriction operator, as is typical in the multi-grid framework [12]. After processing, deeper levels are expanded and the results added back in to the level above: this expansion is by the adjunct prolongation operator. Some additional noise reduction is achieved by repeated application of the bitonic at this level, but with $t = t_{n+1}$, i.e. the threshold from the lower level. The restriction and prolongation operations, denoted by diagonal arrows in Fig. 1, are not lossless: to account for this, a reduced and immediately expanded image is subtracted from the original before adding in the lower level results.

III. METHODS

The key novelty in this paper is the development of masks $w(x)$ which have a completely unconstrained shape at each pixel $x$, and consequent adjustments to the overall bitonic framework which also allow application to images with real noise. The masks control which pixels are included in the forward and inverse ranking operations of Eqs. (1) and (2) and hence fundamentally change the output of the overall bitonic
Fig. 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 \( t_{n+1} \) from the next level down.

![Multi-resolution diagram](image)

Fig. 2. Mask selection for bitonic filters. A section of a noisy image (a) from the SIDD data set [3] has been filtered by (b) the original bitonic filter [6], (c) the structurally-varying (V) bitonic filter [7] and (d) the novel locally-adaptive (X) bitonic filter presented in this paper. The major difference is the way in which the mask \( w(x) \) at each pixel \( x \) is chosen. This is fixed to a circle for the original bitonic filter, can be chosen from a small set of ellipses with varying aspect ratio and orientation for Bitonic V, and is fully adaptive as described in Section III-A for the new Bitonic X. For easier comparison, the masks (bottom left, for the pixel outlined in red) all have the same region size \( l \times l \): a smaller circle would be more appropriate for (b).

![Mask selection diagrams](image)

The locally-adaptive mask domain is designed in Section III-A and application to images with real noise is developed in Section III-B. Consideration is then given to the key threshold parameter in Section III-C and how it is adjusted for each level in the multi-resolution framework in Section III-D. Efficient implementation of the locally-adaptive ranking operations is also explored in Section III-E.

### A. Locally-adaptive masks

In [7] the mask shape \( w(x) \) was allowed to vary, but only 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. A more flexible local mask which can adapt to the local signal shape, but not the noise, would dramatically increase the performance of the bitonic filter. However, care is needed to find ways to exclude the effects of noise on this shape. In this paper, adaptability is increased by removing the elliptical constraint: the mask at \( x \) can potentially include any element in the \( l \times l \) region. Resistance to noise is achieved by making use of several carefully designed thresholds across different image domains and only including elements in the mask when all these domains agree. This new filter is called Bitonic X (or MX for the multi-resolution version).

Figure 3 shows the image domains and defines the terms used in creating the threshold for each domain. The 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. Grey-scale data smoothed with the anisotropic Gaussian of Eq. (10) 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 a global threshold \( t \) which is set depending on the overall noise level in the image: see Section III-C for how this is initially estimated and Section III-D for how it is adjusted for each deeper level in the multi-resolution framework. An element \( y \) in Eq. (7). This difference in mask definition between the existing and new bitonic filters is summarized in Fig. 2, where elements in the mask for a particular pixel (in red) are shown as full grey squares within the overall \( l \times l \) mask region. Each element \( y \) in the mask \( w(x) \) is a vector from the central pixel \( x \) to one of these squares. Smaller grey squares represent elements that might be included at other pixels.
is included in the local mask $w(x)$ only if $|I(x) - I(x + y)|$ is below the appropriate threshold for all of the three domains.

The global threshold $t$ is first adjusted by a factor $f$ which allows local variation over the image:

$$f' = \begin{cases} f_{\text{low}} & r_g > r_{\text{high}} \vspace{1mm} \\ f_{\text{high}}/(r_{\text{high}} - t) + f_{\text{low}}(t - 1)/(r_{\text{high}} - t) & r_g < t \vspace{1mm} \\ 1 - f_{\text{low}} (1 - f') & (f' < 1) \& (m_g < f_{\text{low}}) \vspace{1mm} \\ f' & \text{otherwise} \end{cases}$$

where $m_g$ and $r_g$ vary with $x$ and depend on the noisy grey-scale statistics, as shown in Fig. 3. $f_{\text{low}}$, $f_{\text{high}}$ and $r_{\text{high}}$ are constants which are not expected to change at all for different image types; they were designed based on the performance of a small set of images within the AWGN set described in Section IV-A and remained constant for all other images and datasets.

The factor $f'$ allows a range of local thresholds from $f_{\text{low}} t$ to $f_{\text{high}} t$. The lower limit is appropriate when the local range of data values is high compared to the noise level. 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 $x$. 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 (12) 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 $x$ is sufficiently close to the local minimum or maximum, given the noise level, to indicate it is biased away from the mean. Equation (13) 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:

$$t_g = \begin{cases} 1/\sqrt{3} f t & \text{if } f' \leq 1 \\ \sqrt{3} f t & \text{if } f' > 1 \end{cases}$$

where $t_g$ is set according to the distance from the nearest extremum $m_g$. 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 III-D, 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 actual threshold $t_{sg}$ used with the smoothed grey-scale data needs further adjustment to account for the possibility that the signal in this data has become corrupted. In that case, the value $v_{sg}$ at $x$ may no longer be from the same distribution as the noisy grey-scale value $v_g$ at $x$. If so, these values would
be further apart than would be expected given the noise levels in the noisy and smoothed grey-scale images:

\[
\begin{align*}
    r_v &= |v_g - v_{bg}| \\
    r_t &= \frac{1}{2} \left( t_g + t'_g \right) \\
    t_{bg} &= \begin{cases} 
        \frac{t'_g}{\text{erf}(0.25)} \\
        \frac{t_g}{\text{erf}(0.25)} + 2 (r_v - r_t)
    \end{cases} \text{ for } r_v < r_t \\
    \text{otherwise}
\end{align*}
\]

(17)

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

One further consideration for \(t_{bg}\) is that, for images with low noise and hence requiring shorter 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 \(l\), \(t_{bg}\) 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 \(t_{bg}\) at \(x\) from the global threshold \(t\), elements are included in \(w(x)\) where \(I(x+y)\) is within the appropriate threshold of \(I(x)\), for all relevant \(y \in w_1\) (a rectangular window of size \(l \times l\)), and all three image domains of \(I(x)\). Figure 3 is an example with fairly high noise, and Fig. 4 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 \(x\), but the same set of masks \(w(x)\) are used for all colour channels to prevent colour separation in the final image.

B. 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’ noise) with only a small additive Gaussian component (‘read’ noise) [13]. The sensors are overlayed with a colour filter array, where each \(2 \times 2\) 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 [14], [15]. 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 [16]. 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 \(B_r, B_g, B_b\) and \(B_{bg}\) 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. (14) 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 \(l\) [17].

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 [18]. This has no effect on the various ranking operations, since gamma correction is order-preserving, but it does bias the smoothing filter \(G_{r,\alpha}\). 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.

C. Selection of global threshold

For AWGN noise with known variance \(\sigma_n^2\), the global threshold at the first level, \(t_1\), should be set to \(4\sigma_n\), except for low noise above a signal-to-noise level of \(\approx 22\) dB. In this case a smaller global value of \(t_1\) no lower than \(2.5\sigma_n\) preserves the signal better, and \(f'\) is subsequently set to 1 in Eq. (12) for all levels since the threshold has already been reduced. Given the constants in Eq. (11), this limits the local threshold to between \(2.5\sigma_n\), \(6\sigma_n\).

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) [19]. Non-AWGN noise in images can be modelled as having a variance which is a function of intensity, in which case similar techniques are employed to estimate the Noise Level Function (NLF) rather than a single variance [20], [21]. 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 [22].
Images of \(\sigma_n\) for mobile phone data are hence estimated by first creating \(2 \times\) sub-sampled images, then taking the absolute 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 \(2 \times 2\) Bayer block is kept to account for differences in variance over each colour channel, then scaled by 2.34 to convert to \(\sigma_n\) values.

A variance stabilising transform [23] 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 \(t_1\) of \(5.5\sigma_n\) was used for these images, where \(\sigma_n^2\) was the average variance seen in the data.

\subsection*{D. Multi-resolution thresholds}

The threshold \(t_1\) for the first level in the multi-resolution framework is set according to the image noise, as described in the previous section. Subsequent thresholds \(t_n\) for \(n > 1\) are calculated from the expected reduction in noise due to the bitonic filter applied at the previous level. The reduction after the first level will be inversely proportional to filter length \(l\), since the noise variance reduces approximately with the size of the mask. However, for subsequent levels the reduction is by a factor of two:

\[
\begin{align*}
t_n &= \begin{cases} 
\frac{2}{\sqrt{n!}} t_1 & n = 2 \\
\frac{1}{2} t_{n-1} & n > 2
\end{cases}
\end{align*}
\tag{18}
\]

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).

\subsection*{E. Efficient ranking implementation}

Ranking operations with fixed masks can be implemented extremely efficiently in nearly constant time with respect to the mask width \(l\) [24] with either direct sorting or histogram-based approaches. However, if the mask \(w(x)\) is different at each location, the sorting required for forward ranking in Eq. (1) either needs to be calculated afresh at each \(x\), or a sorted list of the superset of possible mask elements covering \(l \times l\) updated more efficiently, and the required elements extracted 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 \(x\) to all pixels \(x + y\) for \(y \in w(x)\). A sorted list is updated at each \(x + y\) 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 \(\approx 2 \times 0.08 \times I^2 \times |I|\) 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 \(z\) as soon as the forward mask \(x\)
has moved beyond the point where any further update to the sorted list at $z$ is possible.

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

IV. RESULTS

A. Images with AWGN

The fixed-shape bitonic was compared to various linear and morphological filters in [6], with clear improvement relative to all other morphological-based filters, e.g. the OCCO filter [25], and self-dual filters [26], [27], [28], [29]. The structurally-varying Bitonic V was demonstrated in [7] to outperform Non-Local Means [30], anisotropic diffusion [31] and image-guided or bilateral filters [32], [33], but not Block-matching [8], 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 [7], but with a greater range of signal-to-noise ratios (SNR) from $\approx 36$ 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 $\sigma^2_n$ is the added noise variance, is:

- **BM3D** Block-matching\(^2\) [8], with $\sigma$ 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\(^3\) [30], with varying $l$ for window and search length, and the parameter $h$ set to $\sigma_n$.
- **Diffusion** Anisotropic diffusion [31], using a fast MATLAB implementation\(^4\), with iterations $n = \frac{\sigma}{2}$, integration constant set to $\sigma_n$, 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 [6], with varying mask diameter $l$, $c = 0.10$ and $t_1$ set to $2.8\sigma_n$.
- **Bitonic V** Structurally varying bitonic filter, as in [7], with varying mask width $l$, $c_1 = 4\%$, $\alpha = 0.6$ and $t_1$ set to $2.8\sigma_n$.

1. including http://decsai.ugr.es/cvg/CGBase.htm
2. MATLAB BM3D v2.0 software from http://www.cs.tut.fi/~foi/GCF-BM3D/

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 $t_1$ set to $4\sigma_n$ (except for the lowest noise levels, where $t_1$ approached $2.5\sigma_n$).

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

Median X A median filter with the same spatially-varying masks $w(x)$ 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. 5, with examples in Fig. 6, where $\sigma_n$ is expressed relative to the maximum range of the original image data.

B. Images with real noise

The Smartphone Image Denoising Dataset (SIDD) [3] and the Darmstadt Noise Dataset (DND) [5] 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) [34] contains DLSR-like images only, as does the slightly older Renoir dataset [4].

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 $256 \times 256$ 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 [3], with further results from a recent competition in [35].

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 $512 \times 512$ 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 [5], with many additional results, together with relevant citations, on the associated website.

Additional filters tested on these images follow the discussion in Section III-B:

- **G** A version of the filter which includes inverse-gamma (i.e. the removal of standard sRGB
Best filter length
SNR relative to Gaussian / dB
10
25
30
35
-2
-1
5
2
3
4
5
-10 0 10 20 30 Input SNR / dB
0.2 0.4 0.6 0.8 1 Input SSIM
Gaussian Guided Diffusion NLM BM3D Anisotropic Median: X Bitonic Bitonic V Bitonic MV Bitonic X Bitonic MX
(a) SNR and SSIM performance
(b) Filter length and processing time
Fig. 5. 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.

V. DISCUSSION

The AWGN results for non-learning-based filters, which are comparable to those from the previous paper [7], are considered first. It is clear from Fig. 5 that 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 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. 6(a) and (d)) 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. There are some occasions where Anisotropic, Median X or Bitonic X perform well on their own, but the summary results are considerably worse than for Bitonic MX. It is, however, notable that 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, Bitonic MX is better than would have been expected given the performance across AWGN noise. Despite the real image noise in both the SIDD and DND data sets being 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. 5 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).
<table>
<thead>
<tr>
<th>SNR, SSIM</th>
<th>Noise</th>
<th>Median X</th>
<th>Bitonic MV</th>
<th>Bitonic MX</th>
<th>BM3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Original</td>
<td>-7.46 dB, 0.001</td>
<td>16.43 dB, 0.395</td>
<td>18.55 dB, 0.713</td>
<td>19.91 dB, 0.756</td>
<td>19.25 dB, 0.720</td>
</tr>
<tr>
<td>(b) Original</td>
<td>-2.09 dB, 0.016</td>
<td>16.47 dB, 0.488</td>
<td>17.42 dB, 0.594</td>
<td>18.16 dB, 0.625</td>
<td>18.02 dB, 0.621</td>
</tr>
<tr>
<td>(c) Original</td>
<td>6.22 dB, 0.051</td>
<td>22.37 dB, 0.659</td>
<td>22.88 dB, 0.715</td>
<td>23.77 dB, 0.736</td>
<td>23.66 dB, 0.740</td>
</tr>
<tr>
<td>(d) Original</td>
<td>4.26 dB, 0.155</td>
<td>15.85 dB, 0.659</td>
<td>14.73 dB, 0.628</td>
<td>16.38 dB, 0.707</td>
<td>16.83 dB, 0.719</td>
</tr>
<tr>
<td>(e) Original</td>
<td>4.90 dB, 0.149</td>
<td>16.40 dB, 0.558</td>
<td>15.66 dB, 0.506</td>
<td>16.83 dB, 0.578</td>
<td>17.53 dB, 0.636</td>
</tr>
<tr>
<td>(f) Original</td>
<td>-8.80 dB, 0.007</td>
<td>11.44 dB, 0.321</td>
<td>12.18 dB, 0.522</td>
<td>12.60 dB, 0.548</td>
<td>12.49 dB, 0.538</td>
</tr>
</tbody>
</table>

Fig. 6. Results on AWGN images (a) ‘south sound’ with $\sigma_n = 1.28$, (b) ‘peppers’ with $\sigma_n = 0.64$, (c) ‘fruits’ with $\sigma_n = 0.32$, (d) ‘del presepe’ with $\sigma_n = 0.32$, (e) ‘louvre’ with $\sigma_n = 0.32$ and (f) ‘marina bay’ with $\sigma_n = 1.28$. 
Hence there are potentially larger groups of pixels dominated by noise, which 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 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 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
<table>
<thead>
<tr>
<th>Block</th>
<th>Original SNR, SSIM</th>
<th>Noise SNR, SSIM</th>
<th>BM3D G SNR, SSIM</th>
<th>Bitonic MX SNR, SSIM</th>
<th>Bitonic MX Raw SNR, SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>7.79 dB, 0.265</td>
<td></td>
<td>21.21 dB, 0.899</td>
<td>21.97 dB, 0.908</td>
<td>22.08 dB, 0.904</td>
</tr>
<tr>
<td>(b)</td>
<td>8.84 dB, 0.131</td>
<td></td>
<td>24.59 dB, 0.840</td>
<td>25.22 dB, 0.854</td>
<td>25.51 dB, 0.856</td>
</tr>
<tr>
<td>(c)</td>
<td>6.58 dB, 0.296</td>
<td></td>
<td>20.00 dB, 0.906</td>
<td>20.83 dB, 0.915</td>
<td>22.52 dB, 0.932</td>
</tr>
<tr>
<td>(d)</td>
<td>8.59 dB, 0.378</td>
<td></td>
<td>23.21 dB, 0.933</td>
<td>24.52 dB, 0.947</td>
<td>24.64 dB, 0.945</td>
</tr>
<tr>
<td>(e)</td>
<td>6.82 dB, 0.206</td>
<td></td>
<td>18.70 dB, 0.814</td>
<td>18.81 dB, 0.815</td>
<td>19.19 dB, 0.815</td>
</tr>
<tr>
<td>(f)</td>
<td>5.17 dB, 0.135</td>
<td></td>
<td>20.19 dB, 0.846</td>
<td>21.74 dB, 0.870</td>
<td>22.09 dB, 0.870</td>
</tr>
</tbody>
</table>

Fig. 8. Results on sample SIDD images (blocks) (a) 6 (5), (b) 11 (11), (c) 28 (14), (d) 19 (2), (e) 19 (29) and (f) 33 (13).
Fig. 9. Results on sample DND images (blocks) (a) 1 (18), (b) 16 (9) and (c) 26 (2).
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.

It is also worth looking at residuals (that is, the difference between the processed image and the original), as shown in Fig. 10 for two images with particularly high levels of noise. It is clear from these images that there is signal loss for both BM3D and Bitonic MX, hence a ‘ghost’ of the original image is still apparent in the residual. However, in areas where the original image is relatively smooth and constant, Bitonic MX is also fairly constant, whereas BM3D introduces ‘false’ features which could, in some cases, be mistaken for real signals. In this sense Bitonic MX is better behaved than BM3D for very high noise, though this is only a subjective opinion.

The new filter is a clear improvement over other non-learning-based filters on the SIDD and DND data sets, as well as over the more recent WNNM filter, which reported better performance than BM3D on AWGN data. The performance improvement over the TWSC filter, which was specifically designed for real image noise, is less dramatic. It is worth noting, however, that results for TWSC are only available for the lower noise DND data. It is possible that the gap in performance would be larger for the SIDD data, following the trend as for BM3D. In any case, both the WNNM and TWSC filters are considerably slower than either Bitonic MX or BM3D.

Learning-based approaches can clearly perform better still. CycleISP [37] 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 [35]. Processing times for many of these techniques are either not quoted, or are for massively parallel GPU implementations which are not at all comparable to running on a CPU. But in general, even though training can potentially take very considerable time, many learning-based techniques have subsequent running times which are of the same order of magnitude as for BM3D.

However, how much better such techniques can perform on this data is a fair question. 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 $\sigma_n$ 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 [3], [5] 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 [37], which apply their method and another based on un-processing (UPI) [18], both trained on just the DND data, directly to the SIDD data. As seen in Table I, 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 [40]. The actual benchmark ground truth images are not available, however 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 global threshold at the first level $t_{1}$ and filter size $l$ as potentially tunable parameters. Fig. 5 shows that somewhat smaller filter sizes do tend to generate better results for AWGN data with low noise, but 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 \( \tau \) remains an important parameter, in much the same way as the setting of \( \sigma \) 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 I) resulted in almost identical performance.

VI. 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 after improving the performance of BM3D following the addition of an inverse gamma transformation. Even including the more recent WNNM and TWSC filters, 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 techniques 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


Graham Treece obtained an MA degree in engineering, with distinction, from the University of Cambridge, UK in 1994. Following a few years working as a software engineer in noise cancellation and radio communications, he returned to the University of Cambridge and completed his PhD in 3D ultrasound in 2001. He continued to pursue ultrasound research at Cambridge during a five year research fellowship from the Royal Academy of Engineering, focusing on the measurement of attenuation and elastography. From 2008 to 2015 he was the inaugural Evelyn Trust lecturer in Engineering for Clinical Practice in Cambridge, with research spanning most medical imaging modalities. His recent research has mainly concerned X-ray Computed Tomography, developing Cortical Bone Mapping (CBM) for the detailed analysis and visualisation of bone diseases, and also Refined Metal Artefact Reduction (RMAR) for reducing noise due to the presence of metal. He is currently a professor of information engineering at the University of Cambridge. He is an author of a variety of free software packages, including Stradview (medical data visualisation and CBM) and wxDicom (image processing and RMAR).