Morphology-based noise reduction: structural variation and thresholding in the Bitonic Filter

Graham Treece

CUED/F-INFENG/TR 705

August 2018

Cambridge University Engineering Department
Trumpington Street
Cambridge CB2 1PZ
England

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

Abstract

The bitonic filter was recently developed to embody the novel concept of using signal bitonicity to differentiate from noise. This is a combined linear/morphological operator, and the use of data ranking leads to good noise-reduction performance across smooth and disjoint signals alike. However, for processing images, the spatial extent of the bitonic filter was locally constrained to a fixed circular mask. Since natural images tend to have structure which varies over the image, a novel structurally varying bitonic filter is presented, in which the mask is locally adaptive to the signal in the image, but does not follow patterns in the noise. This new filter includes novel robust structurally varying morphological operations, for which an efficient implementation is also developed, and a novel formulation of non-iterative directional Gaussian filtering. Data thresholds are also integrated with the morphological operations, increasing noise reduction for low noise levels, and enabling the filter to be embodied in a multi-resolution framework for better performance at high noise levels. The structurally varying bitonic filter is presented in detail without presuming prior knowledge of morphological filtering, and compared to a number of high-performance linear noise-reduction filters, to set this novel concept in context. These are all tested over a very wide range of noise levels, on a fairly broad set of images, using conventional performance measures. The new filter is a considerable improvement on the fixed-mask bitonic, outperforms anisotropic diffusion and image-guided filtering at all but extremely low noise levels, non-local means at all noise levels, but not the block-matching 3D filter, except for very high noise. Nevertheless, the structurally varying bitonic tends to have less characteristic residual noise in regions of smooth signal, and very good preservation of signal edges, though with some loss of small scale detail when compared to the block-matching 3D filter. The efficient implementation means that processing time, though slower than the fixed-mask bitonic filter, remains competitive.

1 Introduction

There are many situations in which a corrupting additive component (noise) needs to be removed from an *a priori* unknown digital signal, for instance a two-dimensional image. Whilst this problem has no universal solution, natural and synthetic images have features which are sufficiently different from typical noise to enable the development of many practically useful noise-reduction algorithms. The performance and characteristics of such algorithms are critically dependent on how they each define the difference between 'noise' and 'signal'. Such definitions are wide-ranging, for instance based on transformed domains (noise has higher frequency), data level (signal has greater difference in data value), pattern-matching (signal structure is more repeatable) or even the historical record (signal is previously known).

In this context, the novel concept of using signal 'bitonicity' for differentiation was recently proposed (Treece, 2016), where the signal is deemed to be anything containing one maximum or one minimum over a given spatial range. This definition is crucially independent of data value, being based instead on data *ordering* or *rank*: hence it applies equally to both smooth and disjoint signals. The bitonic filter was developed as a combination of rank-based (robust morphological openings and closings) and linear (Gaussian filtering) operators, in order to remove non-bitonicity (noise) from corrupted signals, with an initial application in medical computed tomography (Treece, 2017). It was shown to have good noise-reduction performance across a range of noise levels, surpassing other morphology-based alternatives, and even competing in some cases with more well known linear filters such as anisotropic diffusion (Weickert, 1998) or non-local means (Buades et al., 2005), particularly for images with varying noise levels.

Whilst, for one-dimensional signals, the spatial range over which bitonicity is imposed is defined purely by the filter window length, for two-dimensional (2D) images, a 2D structuring element or 'mask' defines which image pixels locally contribute to the data ranking. In the bitonic filter, this mask was fixed over an image to a circle with a chosen diameter. Since the mask shape imposes a structure on the signal, and yet the structure of the signal is not expected to be constant over an image, it is clear that allowing the shape to vary could offer significant performance improvements.

This concept of structurally varying morphological operations (also known as adaptive morphology) has been the subject of recent research (Ćurić et al., 2014) with well developed mathematical foundations (Bouaynaya and Schonfeld, 2008). There are a variety of ways to change the shape of the mask, and Ćurić et al. (2014) helpfully note that the extent to which the mask is allowed to shape itself to the data is related to the noise reduction performance. Strong prior definitions of shape are required if high noise reduction is desired, since otherwise the mask can conform to the noise as well as the signal. Pre-defined flat masks (Landström and Thurley, 2013) are hence the most appropriate for inclusion in the bitonic filter, with ellipses the simplest extension from a circle. Having defined the type of structurally varying mask, its orientation and specific shape needs to be able to adapt to the image data. The orientation is usually derived from the gradient, often in the form of a structure tensor (Landström and Thurley, 2013; Soille and Talbot, 2001; Tankyevych et al., 2009) also called the average squared gradient (Verd-Monedero et al., 2011). The local anisotropy can be used to control the shape, for instance setting the aspect ratio of an ellipse (Landström and Thurley, 2013).

It is the aim of this paper to develop the bitonic filter so that it can make use of structurally varying masks. Structural variation is a relatively new development within mathematical morphology, and, other than the median filter, mathematical morphology itself is a less widely used technique than linear image filtering, perhaps due to complexity in implementation and analysis. Hence the paper covers sufficient detail to show how the structurally varying bitonic can be implemented efficiently, without presuming familiarity with the field. In order to develop this filter, several extensions to structurally varying morphology are also proposed. Firstly, the bitonic filter unusually involves robust (not involving maxima or minima) morphological openings and closings, and efficient implementations of such operations do not exist. In addition, a better definition of mask orientation is developed, making use of trials over multiple masks rather than solely relying on the structure tensor. The incorporation of data thresholds, and the inclusion of the whole in a multi-resolution framework, is also demonstrated. For the new bitonic filter, an alternative to the Gaussian filter which can structurally match the varying morphological operations is also required.

Since the bitonic filter is a recent concept, its performance is placed in context by comparison with a wide range of more well known linear filters, on a range of images, across multiple levels of noise. It is seen that the structurally varying bitonic can achieve competitive noise reduction, with practical processing times, across the entire noise range, though with particularly promising results in very high noise.

2 Methods

The bitonic filter was analysed in detail in (Treece, 2016). Briefly, it consists of a robust opening $O_{w,c}$ and closing $C_{w,c}$ of a signal $I(\mathbf{x})$ at location $\mathbf{x}$:

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

$$O_{w,c} = r_{w,100-c}(r_{w,c}(I(\mathbf{x}))) \quad (2)$$

$$C_{w,c} = r_{w,c}(r_{w,100-c}(I(\mathbf{x}))) \quad (3)$$

where $r_{w,c}$ is a rank filter, w is the filter region (or mask in 2D), $|w|$ is the window size (or number of elements in 2D) and c the chosen centile, usually set to 10% for a bitonic filter with a fixed circular mask. 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.

Opening and closing operations are not self-dual (symmetric in data value), and do not preserve mean signal values. To correct this defect, the operations are weighted, by considering their difference from the original signal. This difference is filtered with a Gaussian linear filter, $G_\sigma(\mathbf{x})$, with standard deviation $\sigma = 0.33l$ where l is the diameter of the mask in 2D:

$$\epsilon_O = |G_\sigma(I(\mathbf{x}) - O_{w,c})| \quad (4)$$

$$\epsilon_C = |G_\sigma(C_{w,c} - I(\mathbf{x}))| \quad (5)$$

$$b_{w,c} = \frac{\epsilon_O C_{w,c} + \epsilon_C O_{w,c}}{\epsilon_O + \epsilon_C} \quad (6)$$

where ϵ_O and ϵ_C are smoothed opening and closing errors, and $b_{w,c}$ is the output of the bitonic filter. The opening and closing operations effectively detect bitonic signals, which are removed from the original signal, leaving the Gaussian to reduce any residual noise. This preserves bitonic signals, but reduces noise in all regions, including across signal edges.

A slightly different formulation of eq. (6) is possible:

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

where n is a positive integer. $n = 3$ slightly improves the performance, and is used in the remainder of this paper: $n = 1$ would result in the same expression as eq. (6). A typical result of this bitonic filter is shown in Fig. 1(c) for the noisy image in Fig. 1(a). For comparison, the result of applying $G_\sigma(I(\mathbf{x}))$ alone is in Fig. 1(b).

The following sections cover the extension of eq. (7) to the structurally varying (SV) version. Section 2.1 discusses the choice of masks, with appropriate centiles, and the efficient implementation of robust SV versions of the openings and closings in eqns. (2) and (3). Section 2.2 considers the incorporation of data thresholds into these operations, and the subsequent extension of eq. (7) to a multi-resolution framework. Section 2.3 develops an appropriate replacement for the Gaussian filters in eqns. (4) and (5) which also exhibits similar structural variation.

Section 2.4 explores better techniques for selecting the mask shape and direction, and also addresses the practical choices when applying such a technique to colour images. Finally, Section 2.5 shows how these can be combined with the filtering and morphological operations in the previous two sections to implement an efficient SV bitonic filter. The effects of each of these stages are summarised in Fig. 1.

2.1 Structurally varying robust morphological operations

2.1.1 Design of mask sets

Ellipses of varying orientation and aspect ratio have been used in SV closing and opening operations before (Landström and Thurley, 2013), with a flexible definition of shape in terms of the two radial parameters. Implementation is improved if the number of masks is restricted, and in any case there is little point using masks which cover nearly identical sets of pixels. More orientations are required for thinner masks, since the percentage overlap is smaller for a given change in orientation. The maximum radius of the masks is a positive integer r, such that the number of pixels along one side of the square region containing all masks is $l = 2r + 1$. More variations in shape (aspect ratio) are possible for larger mask regions l. Hence the number of shapes s_{tot}, aspect ratio a_s and number of orientations $o_{\text{tot},s}$ for shape s, are given by:

$$s_{\text{tot}} = \min\left\{\left\lceil\frac{r}{4}\right\rceil + 1, 2\right\} \quad (8)$$

$$a_s = \frac{1.7 + (r + 0.3)^{\frac{s-1}{s_{\text{tot}}-1}}}{r+2} \quad (9)$$

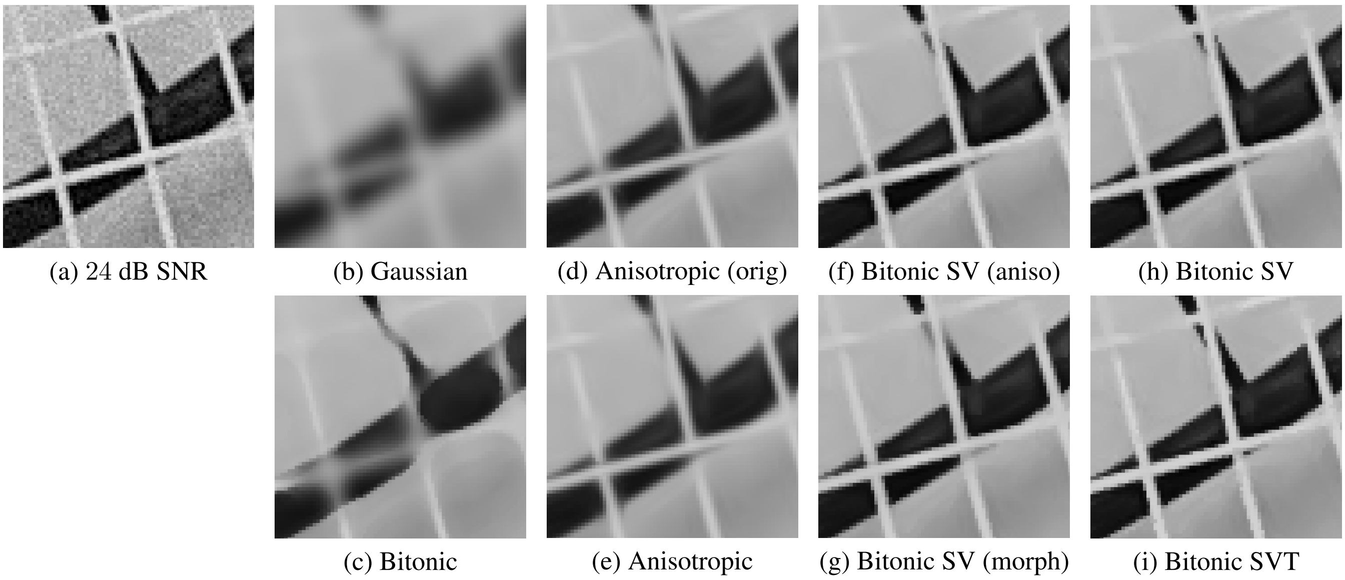
$$o_{\text{tot},s} = \max\left\{4\left[0.4\left(1 - a_s\right)(r+2)\right], 1\right\} \quad (10)$$

Pixels in the $l \times l$ region are deemed to be in each mask if:

$$\left(\frac{r_{\text{major}}}{r + 0.25}\right)^2 + \left(\frac{r_{\text{minor}}}{a_s\,(r + 0.25)}\right)^2 \leq 1 \quad (11)$$

where r_{major} and r_{minor} are the distance of a particular pixel from the centre of the region, respectively along the major and minor axes of the appropriately oriented ellipse. These are both reduced by 0.25 (with a minimum of 0) in order to make the sides of each ‘ellipse’ straighter, particularly for thinner shapes.

The apparently minor details in eqns. (8) to (11) are important, since the exact shape of these masks, for instance the avoidance of single-pixel protrusions, has a noticeable effect on the output of the SV operations. An example for $r = 6$ is given in Fig. 2, in which case $s_{\text{tot}} = 3$, $a_s = \{\frac{1.7}{8}, \frac{4.85}{8}, 1\}$ and $o_{\text{tot},s} = \{12, 4, 1\}$. These masks are never thinner than three pixels, since thinner lines tend to amplify noise characteristics in the image, as has been noted before (Landström and Thurley, 2013).


(a) 24 dB SNR (b) Gaussian (d) Anisotropic (orig) (f) Bitonic SV (aniso) (h) Bitonic SV

(c) Bitonic (e) Anisotropic (g) Bitonic SV (morph) (i) Bitonic SVT

Figure 1: Development of the structurally varying (SV) bitonic filter. (a) Section of a noisy image, filtered in all other images with the same mask size. (b) Gaussian filter, also used in the fixed bitonic (c). (d) Anisotropic filter, with (e) improved corner response, used in the SV bitonic. (f) SV bitonic, with mask selection based on anisotropic filtering, or (g) on morphological operations and (h) a combination of both. (i) SV bitonic with additional data threshold.

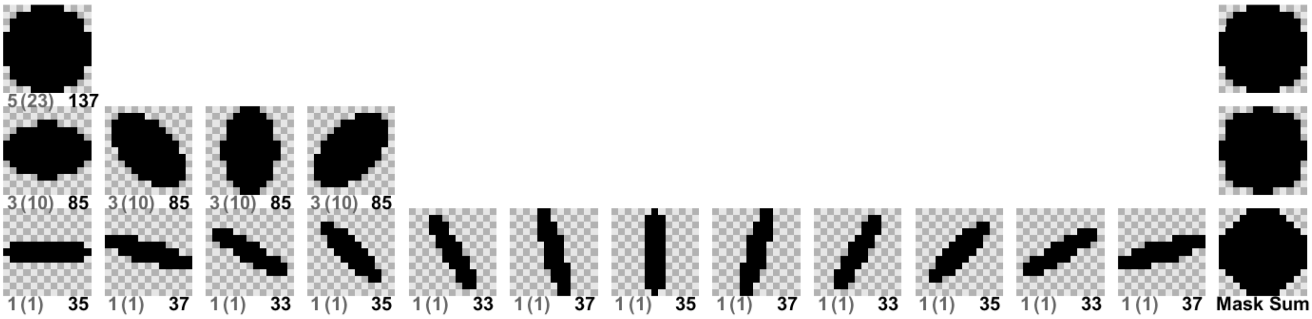


Figure 2: Masks for SV morphology. This is the set of 17 masks for $r = 6, l = 13$, in which case there are three different shapes s (in each row) and up to 12 different orientations (in each column). The numbers below each mask are the rank (from 0) for the equivalent centile c_s (as ranks rather than percentages), followed by the adjusted c_s from eq. (13), followed by the total number of pixels $|w_s|$ in each mask. The three images on the right show the region covered by the sum of the masks in each row.

2.1.2 Equivalent centiles for different mask shapes

For the fixed circular mask, a centile $c = 10\%$ is always a good choice (Treece, 2016). For SV masks with thinner shapes, any of the corresponding orientations could be chosen for processing a particular pixel in an image. As shown in Section 2.4, one way to select the appropriate orientation is based on the output of $r_{w,c}$ in eq. (1), i.e. that which best fits the data. In an image which only contains noise, there is no underlying structure and hence there should be no preference between mask shapes. However, if all masks used the same centiles, the thinner masks would seem to fit the data better simply because there are more orientations to try: effectively these thin masks would over-fit to apparent structure in the noise.

For a given set of masks applied to pure noise, it can be calculated experimentally which centile for each fatter shape (larger s) returns the same ranked value (on average) as for the thinnest shape, presuming the best-fit orientation is used each time. This relationship is a function of the centile c_1 of the thinnest shape, and is also highly dependent on the specific shapes, relative pixel overlap, and number of orientations. However, it is possible to express it as only a function of c_1 by normalisation according to the number of pixels in each mask $|w_s|$, in which case the centile c_s for shape s is:

$$c_s = p(c_1)\frac{1-\sqrt{\frac{|w_1|}{|w_s|}}}{1-\sqrt{\frac{|w_1|+4}{|w_s|}}} \qquad (12)$$

where $p(c_1)$ is a polynomial in c_1 (the centile for the thinnest shape) explaining the remaining experimental relationship. The ‘4’ in the denominator compensates for the discrete shape differences with very small mask sizes. In practice, $c_1 = 4\%$ gives good results in all circumstances, in which case, for mask sets following eqns. (8) to (11), $p(c_1) \approx 6$. The compensated centiles for the specific shapes in Fig. 2 are hence $c_s = \{4, 8.5, 10\}$.

Equation (12) would result in an equal chance of fitting any shape mask if there is no structure in the image, but in fact a fatter mask would be preferred for pure noise, since the larger number of pixels in the mask will improve noise reduction. To ensure this, a positive offset is added to c_s for fatter mask shapes (larger s). This offset needs to be larger for smaller masks (smaller r) since the fewer pixels introduces more variability into the ranking results when applied to noise. c_s is hence adjusted as follows:

$$c_s \quad = \quad c_s + \frac{17}{\sqrt{r}} \frac{s-1}{s_{\text{tot}}-1} \tag{13}$$

This is a well-motivated heuristic, based on experimental results for the noise-ranking variability of masks following eqns. (8) to (11). For the mask set in Fig. 2 it gives $c_s = \{4, 12, 17\}$. The numbers in brackets in Fig.2 show which ranked value these centiles correspond to for each mask.

2.1.3 Histogram-based and sorting-based implementations

Ranking using fixed masks can be performed in nearly constant time with respect to the mask region size l (Perreault and Hébert, 2007), by keeping a sorted list as the mask moves over the image and only updating those values which change between locations. If the data is integer-valued, this list can be stored and updated very efficiently as a histogram of data values, and the rank output at a given centile calculated from this histogram, rather than from a sorted list. For fixed masks, the two ranking operations in each opening or closing can be implemented identically.

For an SV opening or closing, the situation is more complex. In the first or forward pass (inner ranking in eq. (3)), the mask is different at every location. Updating a sorted list is therefore no longer efficient, since there might be a dramatic difference in image pixels covered by each mask between two neighbouring locations. The reverse operation (outer ranking in eq. (3)) is fundamentally different: it is no longer the mask at the current pixel which matters, but whether the masks at the surrounding pixels (used in the forward pass) contain the current pixel. This reverse pass can be performed reasonably efficiently in cases where only the 0% (minimum) and 100% (maximum) centiles are being used, by refining the output in tandem with the forward pass (Tankyevych et al., 2009). Directional filtering, based on masks which are all lines, can also be implemented efficiently (Soille and Talbot, 2001). However, this latter technique does not adapt easily to other shapes, and neither technique is appropriate for robust operations using non-extreme centiles.

Hence a whole new approach is needed for a robust SV opening or closing with a fixed set of pre-determined masks. This approach, summarised in Fig. 3, is based on initially ranking the data using the convex hull of all the masks, and then deducing the results for a particular mask at a given pixel from this superset of sorted data. The convex hull mask can be ranked efficiently since it is fixed, and the additional region index preserves the original location of each sorted value. Any values in the superset of sorted data can then be correctly associated with each mask, and processed appropriately. Preserving the region index is easy for the sorting-based implementation, but has a greater detrimental effect on the histogram-based implementation. Whereas, for fixed masks, the histogram count at each data value was stored and either incremented (when adding a pixel to the sorted list) or decremented (when removing), for SV masks a separate list of region indices must be preserved at each data value, and the correct indices removed.

Figure 4 (a) and (b) contain forward and reverse examples with real data: the small squares in the overlay show the whole superset of sorted data, whilst the larger squares show the data which determines the rank output for the central pixel. It is apparent from this that, whilst the forward operation is recognisable as one of the thin ellipse masks, the reverse operation involves a far more complex mask shape which is related to coverage from the surrounding pixels.

2.2 Combining thresholds with morphological operations

2.2.1 Implementation of relative threshold limits

In noise reduction the variance of the noise is sometimes reasonably well known (or can easily be discovered), and the local range of data values that can be attributed to noise is hence limited by some threshold value. For the best performance in low noise, it is critical to make some use of this to preserve structural discontinuities above this level. Morphological operations do not make any specific use of such information, but the opening and closing operations can be adapted to include it.

The rank output at the correct centile is first selected from the appropriate mask in the forward SV operation. Then, assuming a noise threshold t, if the output at this centile is more than t away from the original data value at the current pixel, then it is replaced by the closest centile in the mask which is within t of the current pixel. This effectively changes the mask shape to ignore pixels whose data values are more different than can be explained by noise, as can be seen in the example of Fig. 4(c). In the reverse operation, pixels are only considered if their forward centile was *not* changed by this threshold, and the rank output at the required reverse centile is also adjusted (as in the forward operation) if it is more than t away from the value of the original image pixel. Once again, this changes the effective mask shape, as seen in Fig. 4(d).

2.2.2 Multi-resolution operation using thresholds

Multi-resolution techniques, which entail processing different versions of an input image at different resolutions, are an obvious way to allow for larger masks without dramatically increasing the processing time. These can involve iterations on the residual image (the original minus the processed version) at a lower resolution, like a Laplacian pyramid (Paris et al., 2011). However the residual contains very little of the image structure on which morphological operations rely, and hence processing of the residual does not make sense for such operations. A similar framework can still be used, but in this case the threshold t_n is lowered with each level n, and applied to the reduced, processed image, rather than the residual. The overall approach is shown in Fig. 5. Reduction of the image to quarter size (half in each dimension) is achieved with a restriction operator, as is typical in the multi-grid framework (Briggs et al., 2000): a Catmull-Rom spline (Catmull and Rom, 1974) is a good choice, since it preserves edges as

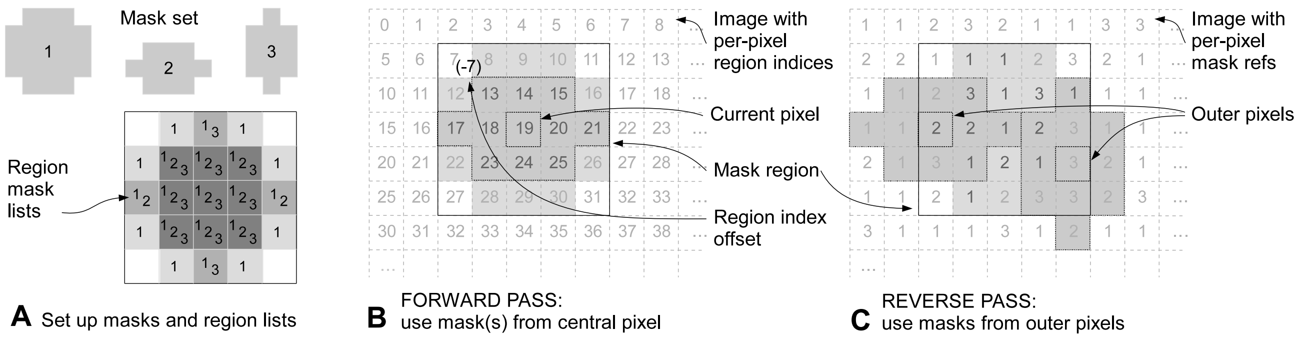

Mask set
1
2
3
Region mask lists
A Set up masks and region lists
Image with per-pixel region indices
Current pixel
Mask region
Region index offset
(-7)
B FORWARD PASS: use mask(s) from central pixel
Image with per-pixel mask refs
Outer pixels
C REVERSE PASS: use masks from outer pixels

Figure 3: Implementation of robust SV opening and closing. **A** Each mask is given a number (1 to 3 in the simple example above), and an array of width l, a square region bounding all the masks, is initialised with a list of mask intersections at each pixel. **B** Morphological operations proceed using the (fixed) convex hull of all these masks (shown in light grey), which leads to very efficient ranking of all the image values within this area. An additional index is stored for each pixel, equal to $i \times l + j$, where (i, j) is the zero-based image pixel location. This is retained in the ranked list of image values, and preserves the location of the pixel even after ranking: subtracting the index from the top-left of the current region provides the location of each image value relative to the region mask lists. Hence the output of the forward ranking operation for different masks, with different centiles, can be calculated from the ranked superset. **C** For the reverse operation, ranking again starts using the convex hull of the masks. The combination of the region index, and the per-image-pixel mask number from the forward operation, can be used to restrict the output to outer pixels from a mask that included the current pixel.

much as possible in the lower-resolution image. Expansion of the image is by an appropriately matched prolongation operator. Both these operators, denoted by diagonal arrows in Fig. 5, are very fast, but the combination is 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 can be chosen according to the expected noise variance, or set to the maximum data value deviation in the image (i.e. 256 for an 8-bit image). Subsequent thresholds t_n are calculated from the expected reduction in noise due to the SV bitonic filter applied at the previous level: this will be inversely proportional to filter length r, since the variance reduces approximately with the size of the mask, hence:

$$t_n \approx \frac{t_{n-1}}{2.4r} \tag{14}$$

where the 2.4 in the denominator accounts for the average SV mask shape and combination of operations in eq. (7). Three levels are typically sufficient, since t_n reduces quite rapidly. These do not add much to the overall processing time, since although the filter radius r is constant (increasing the effective range of the filter at each level), images at lower levels have only one quarter of the pixels. This version of the multi-resolution SV bitonic is referred to as Bitonic MV1.

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 can be achieved, particularly in high noise, by repeat application of the SV bitonic at this level, but with t_{n+1}, i.e. the threshold from the lower level. This option, referred to as Bitonic MV2, will hence tend to double the processing time.

2.3 Linear filtering with structural variation

The fixed bitonic in eq. (7) employs a Gaussian filter to smooth the error and effectively give a mean image value with which to properly adjust the weights for the closing and opening operations. It is important that this filter extends over a similar domain to the morphological operations, so for the SV bitonic, a Gaussian-type filter is needed which will also vary in direction, but over a fixed size domain.

2.3.1 Non-iterative anisotropic filtering

The desired filtering direction can be calculated from the well known structure tensor T, or matrix of smoothed gradients $\{g_i, g_j\}$, in horizontal i and vertical j directions, of $I(\mathbf{x})$:

$$T = \begin{bmatrix} G_\sigma({g_i}^2) & 2G_\sigma(g_i g_j) \\ 2G_\sigma(g_i g_j) & G_\sigma({g_j}^2) \end{bmatrix} \tag{15}$$

$$\equiv \begin{bmatrix} T_{ii} & T_{ij} \\ T_{ij} & T_{jj} \end{bmatrix} \tag{16}$$

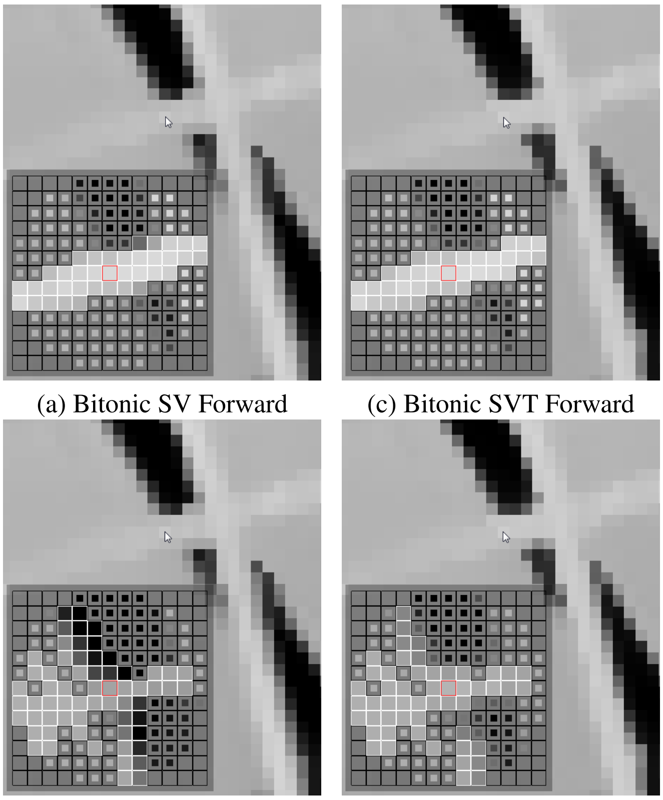
Following Van Vliet and Verbeek (1995), the local direction $\phi(\mathbf{x})$ and degree of anisotropy $\gamma(\mathbf{x})$ can be derived from the eigenvalues $\lambda_{1,2}$ of T:

$$\lambda_{1,2} = \frac{1}{2}\left(T_{ii} + T_{jj} \pm \sqrt{(T_{ii} - T_{jj})^2 + {T_{ij}}^2}\right)$$

$$\gamma(\mathbf{x}) = 1 - \frac{\lambda_2}{\lambda_1} \tag{17}$$

$$\phi(\mathbf{x}) = \frac{1}{2}\tan^{-1}\left(\frac{T_{ij}}{T_{ii} - T_{jj}}\right) \tag{18}$$

where $\gamma = 0$ signifies low anisotropy (no dominant direction) and $\gamma = 1$ signifies high anisotropy (local gradients in only one direction). ϕ is then the angle following the dominant features in the image, i.e. the direction in which the filter should be aligned. The spatial range of γ and ϕ is determined by the extent of the Gaussian filter G_σ in eq. (15), which has



(a) Bitonic SV Forward

(c) Bitonic SVT Forward

(b) Bitonic SV Reverse

(d) Bitonic SVT Reverse

Figure 4: Mask selection and ranking. The images show the result of a robust SV bitonic operation, with the bottom-left overlay containing the mask selection for the current pixel (denoted with an arrow). (a) In the forward operation, the whole region (small squares) is ranked, but only data within the chosen mask (larger squares) considered further. (b) In the reverse operation, data is considered whose masks cover the current pixel. (c) Values beyond a data threshold from the current pixel are discarded after choosing the mask in the forward operation. (d) In the reverse operation, values are only used if they derive from a mask centile which was not affected by such thresholding, and are not themselves outside of the data threshold.

$\sigma = 0.33l$ as before in order to match the mask region size $l \times l$. Figure 6(b) shows $\phi(\mathbf{x})$ for the image in Fig. 6(a) filtered with $l = 13$, and Fig. 6(c) the anisotropy $\gamma(\mathbf{x})$.

γ and ϕ can be used to construct a Gaussian-like filter $G_{\sigma,\alpha}(\mathbf{x})$ which follows the dominant direction, and whose extent changes with anisotropy:

$$d_{\mathbf{y}}(\mathbf{x}) = ||\mathbf{x}-\mathbf{y}|\sin(\angle(\mathbf{x}-\mathbf{y}) - \phi(\mathbf{x}))| \tag{19}$$

$$\Omega_{\mathbf{y}}(\mathbf{x}) = \frac{\mathrm{e}^{-\frac{|\mathbf{x}-\mathbf{y}|^2}{2\sigma^2}}}{\left(\frac{d_{\mathbf{y}}(\mathbf{x})\gamma(\mathbf{x})^2}{\alpha^2}+1\right)\left(\frac{d_{\mathbf{x}}(\mathbf{y})\gamma(\mathbf{y})^2}{\alpha^2}+1\right)} \tag{20}$$

$$G_{\sigma,\alpha} = \frac{\sum_{\mathbf{y}\in w} \Omega_{\mathbf{y}}(\mathbf{x}) I(\mathbf{x}-\mathbf{y})}{\sum_{\mathbf{y}\in w} \Omega_{\mathbf{y}}(\mathbf{x})} \tag{21}$$

where $\mathbf{x}$ is the current pixel location, and $\mathbf{y}$ a neighbouring location within the window w of size $l \times l$. The numerator for the weighting $\Omega_{\mathbf{y}}(\mathbf{x})$, eq. (20), is just a Gaussian in the distance between pixels at $\mathbf{x}$ and $\mathbf{y}$, with standard deviation σ. However, it is reduced by the denominator which takes into account both the anisotropy $\gamma(\mathbf{x})$ and the dominant image direction, via $d_{\mathbf{y}}(\mathbf{x})$. This is the perpendicular distance between the pixel at $\mathbf{y}$ and a line from $\mathbf{x}$ drawn along the filtering direction $\phi(\mathbf{x})$, and hence has a small value if $\mathbf{x}$ and $\mathbf{y}$ lie along the line defined by $\phi(\mathbf{x})$. It is similar, at least in concept, to the geometric weight used in a trilateral filter (Wong et al., 2004). Higher weights are associated with pixels along such a line (low $d_{\mathbf{y}}(\mathbf{x})$) or when the anisotropy is small (low $\gamma(\mathbf{x})$). The existence of both bracketed terms in the denominator ensures that this must also be true for the symmetric case of a line from $\mathbf{y}$ at $\phi(\mathbf{y})$ towards $\mathbf{x}$. It is also easy to introduce a threshold on $|I(\mathbf{x}) - I(\mathbf{y})|$ above which $\Omega_{\mathbf{y}}(\mathbf{x}) = \Omega_{\mathbf{x}}(\mathbf{y}) = 0$.

The overall effect of anisotropy is set by α, which controls the minimum width perpendicular to the main filtering direction when $\gamma(\mathbf{x}) = 1$. Higher values of $\alpha \gg 1$ (or locally lower values of $\gamma(\mathbf{x}) \ll 1$) will lead to $G_{\sigma,\alpha} \Rightarrow G_\sigma$. An example application of $G_{\sigma,\alpha}$ is in Fig. 1(d), in which case $\alpha = 0.6$ and σ is set to the same value as in the isotropic Gaussian in Fig. 1(b). The result is similar to iterative anisotropic diffusion (Weickert, 1998), but the non-iterative implementation allows the range of the filter to be specifically controlled and removes any possibility of instability. It also has some similarity with another tensor-based method (Baghaie and Yu, 2015), though this was used for image interpolation rather than filtering.

If the anisotropy and direction of the Gaussian filter were constant or changed slowly over I, very efficient implementation techniques involving appropriate shearing of the image (Lampert and Wirjadi, 2006) could be used. However, in this case $G_{\sigma,\alpha}$ is $\mathcal{O}(l^2)$, hence less efficient than the separable Gaussian G_σ which is $\mathcal{O}(l)$, but it can still be implemented in reasonable time. The calculation of γ and ϕ is relatively fast since these are filtered with G_σ. Many of the terms in eq. (20) can be pre-calculated across an array covering either the region w or the image I, by judicious use of trigonometric identities, leaving only a small number of simple operations to be performed for each $\{\mathbf{x}, \mathbf{y}\}$ combination. In addition, since $\Omega_{\mathbf{y}}(\mathbf{x}) \equiv \Omega_{\mathbf{x}}(\mathbf{y})$, each weighting calculation can be used twice. It may also be possible to speed this up further, following the techniques presented in (Baek and Jacobs, 2010).

2.3.2 Improvement of response at corners

Careful consideration of the example in Fig. 1(d) shows that $G_{\sigma,\alpha}$ introduces a slight diagonal blur where there are strong corners in the image. This is due to the well known problem that the anisotropy γ will be small so long as the eigenvalues of the structure tensor matrix, eq. (15) are the same. This is the desired behaviour in flat regions where they are both small, but is not necessarily beneficial in corners where they can both be large. Hence $\gamma \ll 1$ at the middle of the corner, and also in the surrounding flat region, and the consequent high $\Omega_{\mathbf{y}}(\mathbf{x})$ will cause inappropriate mutual blurring. The prominent diagonal lines of low anisotropy (black) can also be seen across the corners of the tennis racket strings in Fig. 6(c).

This effect can be improved by a slight modification to

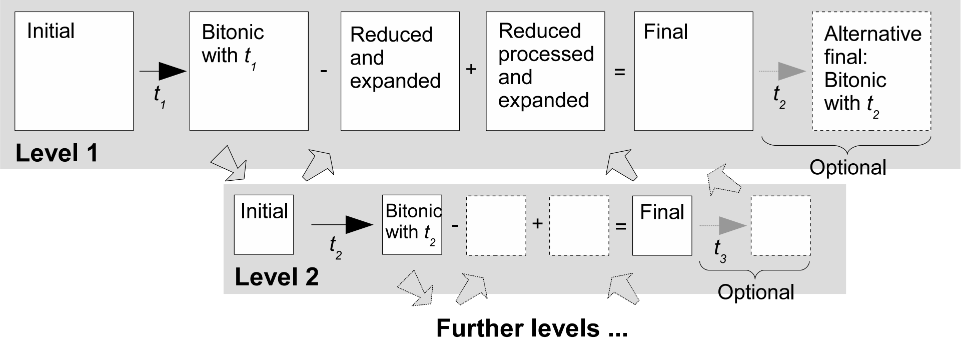
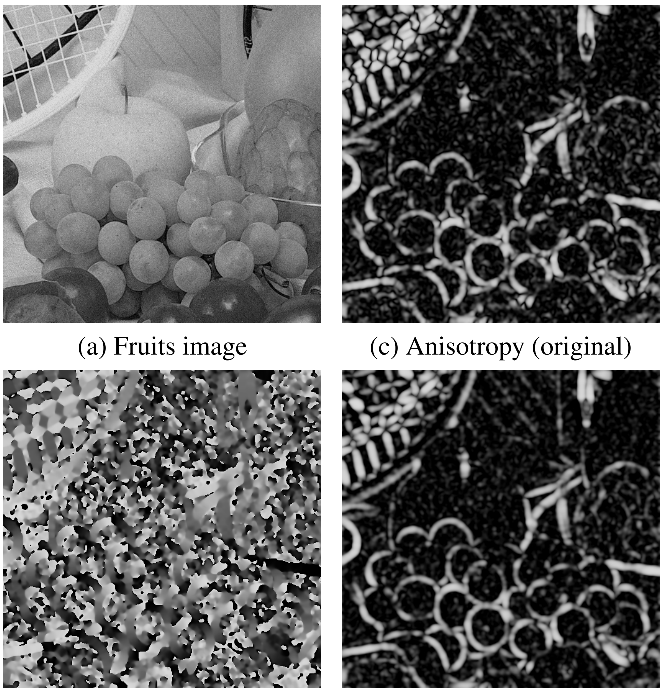


Figure 5: Multi-resolution implementation of the bitonic filter using thresholds. At each level n, the bitonic filter operates on a quarter-size image, with a reduced threshold t_n, but with the same filter radius r in pixels, which means that the effective filter range increases. The threshold is reduced 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). Optionally, the bitonic filter is then re-run at this level, but with an even lower threshold from the next level down.


(a) Fruits image

(c) Anisotropy (original)

(b) Anisotropic orientation

(d) Anisotropy (improved)

Figure 6: Determination of local anisotropy. The extent and direction of anisotropy is shown for the image in (a). (b) The filtering direction is perpendicular to the largest eigenvector of the filtered structure tensor, with vertical as mid-grey and horizontal as white/black. (c) The anisotropy is derived from the difference between the maximum and minimum eigenvalues, with 1 represented as white and 0 as black. (d) Close to the corners, both eigenvalues are large and hence anisotropy is low, despite strong image gradients. This situation can be improved by filtering at different stages: particularly note the more consistent white lines near corners.

eq. (17):

$$\gamma(\mathbf{x}) = 1 - \frac{G_\sigma(\lambda_2)}{\lambda_1} \tag{22}$$

in which the smaller eigenvalue λ_2 is smoothed to give a more stable reference against which to compare the larger eigenvalue λ_1. This removes the inappropriate blurring at corners, as seen in Fig. 1(e), by providing a substantially less noisy anisotropy, but with similar spatial resolution, as in Fig. 6(d).

2.4 Mask shape and orientation

Having defined a robust SV closing and opening, and an appropriately matched linear filter, it remains to choose the particular mask orientation and shape for each pixel in $I(\mathbf{x})$.

2.4.1 Masks from the structure tensor

The structure tensor has been used in SV opening and closing before (Landström and Thurley, 2013), and is already the basis of the linear filtering in Section 2.3, hence it makes sense to consider this technique first. All that is required is to map γ and ϕ to particular shapes and orientations in the set of masks, e.g. in Fig. 2. The mask shape is defined as:

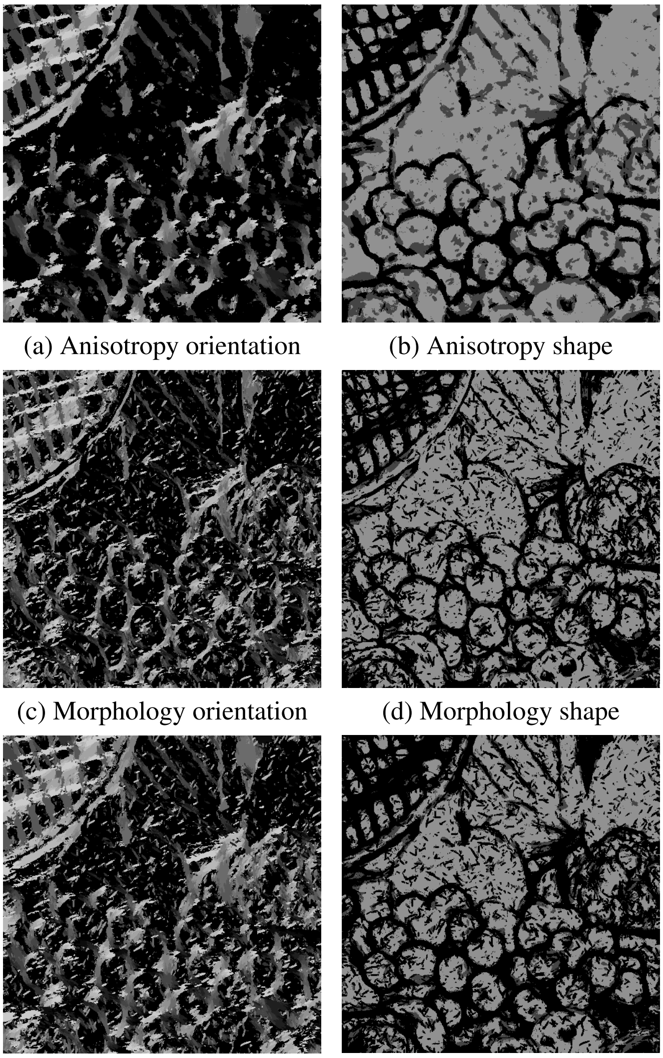
$$s(\mathbf{x}) = \min\left\{\left[\left(1 - \frac{\gamma(\mathbf{x})}{0.8}\right)(s_{\text{tot}} - 1)\right] + 1, 1\right\} \tag{23}$$

where $s(\mathbf{x})$ is the shape number, starting from the thinnest, as in eq. (8). The mask orientation $o(\mathbf{x})$ is simply that from the possible set, for the shape $s(\mathbf{x})$, which most closely matches $\phi(\mathbf{x})$. The resulting orientations and shapes for the masks in Fig. 2 are shown in Fig. 7(a) and (b). The result of applying this technique within an SV bitonic filter is in Fig. 1 (f).

2.4.2 Masks from morphological operations

Since the masks in Section 2.4.1 are based on orientations from the smoothed structure tensor, they are resistant to noise but do not adjust well to small details in the image. An alternative approach is to trial all possible masks during the initial ranking of both the closing and opening operations, and select the mask whose output centile is closest to the median value in the ranked superset of the convex hull of all the masks. This effectively selects whichever mask fits best to the data: and the adjusted centiles in Section 2.1.2 ensure that fatter masks are preferred if there is no apparent structure in the image, even when corrupted by noise.

Whilst trialling all masks may seem prohibitively slow, the implementation in Section 2.1.3 means that a large number of masks can be tested whilst incurring only a small processing overhead. The resulting orientations and shapes for the masks in Fig. 2 are shown in Fig. 7(c) and (d), and the result of this technique within an SV bitonic filter is in Fig. 1(g).



(a) Anisotropy orientation (b) Anisotropy shape

(c) Morphology orientation (d) Morphology shape

(e) Bitonic SV orientation (f) Bitonic SV shape

Figure 7: Mask selection for the SV bitonic filter. To select a mask (in this example from the set in Fig. 2), both the shape (row from Fig. 2) and orientation (column) must be specified. The dominant direction from the structure tensor can determine the mask orientation (a) with anisotropy determining the mask shape (b). Alternatively, the mask type can be chosen from the morphological operations (c) and (d) by trialling several masks and selecting the one which fits the data best. A combination of both approaches is used in the SV bitonic (e) and (f).

2.4.3 Colour images and optimal combination of masks

The morphology-based masks have more detail than those based on information from the structure tensor. However, they are poorly defined at signal edges, since no mask, centred on such points, will fit the data well. Fortunately, these are the locations at which the structure tensor defines the orientation very well. A combination of the two methods is hence achieved by starting with the morphological mask definition, then replacing masks with the structure-tensor definition if this gives a thinner shape (smaller s), so long as the anisotropy γ is greater than $\gamma_{\min}$:

$$\gamma_{\min} = \min\left\{\frac{1.05}{\sqrt{r}}, \frac{1.05}{\sqrt{2}}\right\} \tag{24}$$

which is an experimentally-derived noise threshold on the anisotropy, dependent on the size r of the smoothing filter in the structure tensor. This combination is shown in Fig. 7(e) and (f), with the subsequent result for an SV bitonic filter in Fig. 1(h).

Morphology-based masks can also improve γ and ϕ used in the linear filter of Section 2.3. In this case, eq. (23) is applied in reverse to calculate an effective anisotropy γ_s derived from the morphology-based shape s. If the actual and shape-derived anisotropy are below the expected noise floor, i.e. $\gamma < \gamma_{\min}$ and $\gamma_s < \gamma_{\min}$, then both the anisotropy and orientation are replaced with the shape-derived versions. The linear filter is applied after the SV opening and closing, and hence the masks from the reverse pass in Fig. 3 are used for this purpose.

The simplest way to process colour images is to apply the morphological or linear operations to each colour channel separately. However, using different masks and orientations means that each channel can be smoothed in different directions, resulting in unsightly colour separation. It is hence normally better to define the masks and orientations once from either a grey (for RGB) or lightness (for CIELAB or similar) image, then process all channels individually, but using the same set of masks for all channels. For RGB images, this has the added benefit of reducing the effect of noise on the masks, since the grey image is a reduced-noise average of the RGB channels. In most colour images, structure is largely preserved in the grey version, and the reduction in noise more than compensates for any otherwise small loss of structure.

2.5 Structurally varying bitonic filter

The overall procedure for implementing the SV bitonic filter is as follows:

1. Set the size of the filter r and initial centile for the thinnest shape c_1 (usually 4%). The mask set and centiles follow from eqns. (8) to (13).

2. Calculate $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$ for the image $I(\mathbf{x})$ (or the grey version of $I(\mathbf{x})$ if processing a colour image). Also calculate an optimal set of mask shapes $s(\mathbf{x})$ and orientations $o(\mathbf{x})$ by trialling all possible masks in the initial ranking of the opening and closing operations.

3. Use $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$ to improve the masks in $s(\mathbf{x})$ and $o(\mathbf{x})$, following Section 2.4.3.

4. Perform the robust SV opening and closing operations on each image channel separately, similar to eqns. (2) and (3), using the mask distributions given by $s(\mathbf{x})$ and $o(\mathbf{x})$.

5. Use $s(\mathbf{x})$ and $o(\mathbf{x})$ to improve the anisotropy and orientations in $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$, following Section 2.4.3.

6. Calculate the smoothed errors for each channel, ϵ_O and ϵ_C, with eqns. (4) and (5), but with $G_{\sigma,\alpha}$ instead of G_σ, and the anisotropy and orientations $\gamma(\mathbf{x})$ and $\phi(\mathbf{x})$.

7. The smoothed errors are combined with the SV opening and closing operations using eq. (7) with $n = 3$.

This is referred to as Bitonic SV, with an example in Fig. 1(h).

A data threshold can also be introduced, as in Section 2.2, if the noise level is known or can easily be inferred. This is referred to as Bitonic SVT, with an example in Fig. 1(i). If thresholds are used it is also possible to embed the Bitonic SVT in a multi-resolution framework, as discussed in Section 2.2.2.

3 Results

3.1 Image noise reduction

The fixed bitonic filter was compared to various linear and morphological filters in (Treece, 2016). Improved performance was clearly demonstrated over the morphological filters, including the OCCO filter (Aptoula and Lefevre, 2007), self-dual area-based grain filters (Caselles and Monasse, 2002; Monasse and Guichard, 2000), and self-dual levelling based on reconstruction using a Gaussian mask (Maragos and Evangelopoulos, 2007; Serra et al., 2013). The Bitonic SV is instead compared to a range of high-performance linear filters, selected for the breadth of their approaches to noise reduction. In each case l (maximum diameter of the mask for the bitonic) is used to set the parameter which most controls the extent of the filter:

BM3D Block-matching[1] (Dabov et al., 2007), with the parameter σ set to a variety of trial values, controlled by l, and centred around the actual standard deviation of the added noise. The profile was left at the default setting, i.e. 'normal' ($\sigma < 0.16$), or 'vn' ($\sigma \geq 0.16$).

NLM Non-local means filter, implemented using a fast algorithm for MATLAB[2] (Buades et al., 2005), with the window and search length both set to l, and the filter parameter h set to the standard deviation of the added noise.

Diffusion Anisotropic diffusion (Perona and Malik, 1990), implemented for MATLAB[3], with number of iterations set to l, the integration constant set to the standard deviation of the added noise, the gradient threshold set to twice the standard deviation of the added noise, and the wide-region conduction coefficient.

Guided Image-guided filter, implemented using the MATLAB[4] function imguidedfilter[5] (He et al., 2013), with the local neighbourhood size set to l, and the degree of smoothing set to four times the added noise variance in the image. This is very similar to the well-known bilateral filter (Tomasi and Manduchi, 1998).

Anisotropic The anisotropic Gaussian filter described in section 2.3, with $\sigma = 0.33l$ and $\alpha = 0.6$.

Bitonic Fixed bitonic filter as in eq. (7), with mask diameter l, and $c = 10\%$.

Bitonic T As above, but with added threshold t set to 2.8× the standard deviation of the noise.

Bitonic SV Structurally varying bitonic filter, as described in section 2.5, with $c_1 = 4\%$, $\alpha = 0.6$ and mask region of width l.

Bitonic SVT As above, but with added threshold t set to 2.8× the standard deviation of the noise.

Bitonic MV1 Multi-resolution version of the above, as described in Section 2.2.2.

Bitonic MV2 As above, but including the additional optional filter in Fig. 5.

Where necessary, additional parameters were set presuming knowledge of the added noise, and chosen for optimal signal to noise ratio (SNR) (and, in the case of Diffusion, reasonable stability) at each noise level. However, only l was optimised over individual images and noise levels for the best SNR and SSIM performance in each case, the other parameters being fixed over all images. The data was extended at the image edges, either symmetrically or by repeating the edge value, with similar results in both cases. SNR was calculated using the MATLAB function psnr, whose mean-squared-error is based on the matrix 2-norm.

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

[2] MATLAB file exchange: Fast Non-Local Means 1D, 2D Color and 3D by Dirk-Jan Kroon, 28 Apr 2010

[3] MATLAB file exchange: Anisotropic Diffusion (Perona & Malik) by Daniel Lopes,14 May 2007

[4] MATLAB R2014a, The MathWorks Inc., Natick, MA, 2000

[5] http://uk.mathworks.com/help/images/ref/imguidedfilter.html

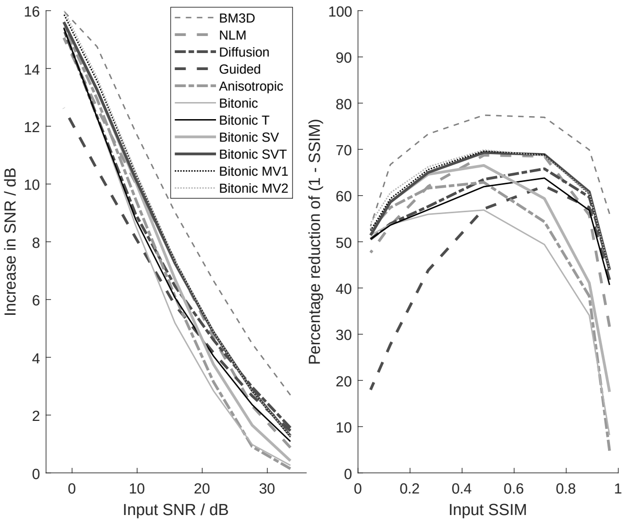


Figure 8: Summarised results over the complete set of 23 images. Optimal filter performance (after optimisation of filter extent l in each case) is averaged over each noise level in each image. Increase in SNR is shown on the left, and percentage reduction of residual SSIM (i.e. the difference from the ideal SSIM value of 1) on the right.

A range of additive Gaussian noise values were tested, giving signal-to-noise ratios (SNR) from ≈ 36 dB (very low noise) to 0 dB (very high noise) in steps of 6 dB, since the relative performance of each filter is expected to vary with noise level. Structural similarity (SSIM) (Wang et al., 2004) was included to evaluate image quality after noise reduction as well as increases in SNR. A fairly large set (given the number of filters, noise levels, and investigated parameters) of 23 images was tested, including standard test images from public-domain sites[6], various high dynamic range (HDR) images all with the CC0 Creative Commons licence, and two simple computer-generated images. Whilst the intention was for these images to cover a fairly broad range of subjects, they can only be properly considered a convenience sample.

Figure 8 summarises the results over all these images, and all of the noise levels. It reveals the characteristic behaviour of each technique, but does not show the variation across different types of image, nor the characteristic residual patterns in the reduced-noise images. Hence Tables 1 and 2 contain detailed results for a selection of these images, with the actual images in Figs. 9 to 12. These figures deliberately cover a wide range of added noise levels, except for very low noise, for which the visual differences are harder to discern.

3.2 Processing times

The processing times for the various algorithms are included in Tables 1 and 2, for whichever value of l generated the best SNR and SSIM results, measured using an i7-7500 CPU at 2.7 GHz. The Bitonic SV algorithm is of particular interest, since both the anisotropic filter $G_{\sigma,\alpha}$ and the implementation of robust SV opening and closing are novel. Figure 17 contains more details on the processing times for these algorithms, all applied to a 512×512 colour image. Fixed and robust SV openings are first considered, both for histogram-based and sorting-based implementations, since whilst the former is more efficient, the latter is necessary if the data is not of integer type. The fixed opening is also implemented in the SV framework, revealing the roughly 3 to 4× overhead purely due to the framework, rather than the increased number of trial masks. Subsequently adding trials of up to 32 masks (for $l = 21$) for the SV version adds a much smaller overhead.

Both implementations of the key parts of the SV bitonic are considered, demonstrating that the overall algorithm is $\mathcal{O}(l^{1.5})$ for the histogram-based version and $\mathcal{O}(l^2)$ for the sorting-based version. Note that both of these performances are acceptable, since l is the filter width, not the number of elements in the filter mask, which is proportional to l^2. All filters in Fig. 17 are based on local windows, and hence also scale with the number of pixels in the signal $I(\mathbf{x})$.

4 Discussion

Development of the SV bitonic filter was motivated by the novelty of differentiating signal and noise via bitonicity, and the apparent gain from allowing the mask shape to vary with the structure of the signal in an image. Figure 8 shows that this is vindicated by the performance of Bitonic SV, which is better than the original Bitonic for both SNR and SSIM, at all noise levels, usually by a considerable margin. This is particularly clear for the boat image in Fig. 9 with low-to-medium noise and for the blue rocks image in Fig. 15 with very high noise. The additional performance does come at an increased, though still competitive, processing time. In the previous work (Treece, 2016), Bitonic was only better than NLM and Diffusion in a few specific cases, most notably for varying noise, whereas Bitonic SV now outperforms both of these linear alternatives for medium to high noise levels, producing distinctly sharper images, as in Fig. 16.

In contrast, the addition of a data threshold to both Bitonic and Bitonic SV has a strong effect on the performance in lower noise scenarios. This is apparent in the numerical results in Tables 1 and 2, but particularly clear from the SSIM results in Fig. 8. Both Bitonic T and Bitonic SVT offer much better SSIM improvement at low noise than their non-thresholded counterparts, but tend to the same performance at higher noise. Bitonic SVT outperforms NLM at all noise levels, and generates less distracting characteristic patterns. This is clear in the synthetic shapes image in Fig. 10 but is also apparent in natural images, for instance the monarch in Fig. 14. In these examples Diffusion also generates a visually pleasing result, though both this and Guided fail to do so at most other noise levels. Guided is, however, extremely fast, and does have good performance at the lowest noise levels.

The improvements in Bitonic SVT are not, however, sufficient to achieve better SNR and SSIM results than BM3D, which is the strongest performer, except at the highest noise levels. Here, the addition of the Bitonic MV2 multi-resolution framework to Bitonic SVT allows it to outperform even BM3D (see Tables 1 and 2), though at the cost of doubling the processing time. In contrast, Bitonic MV1 offers

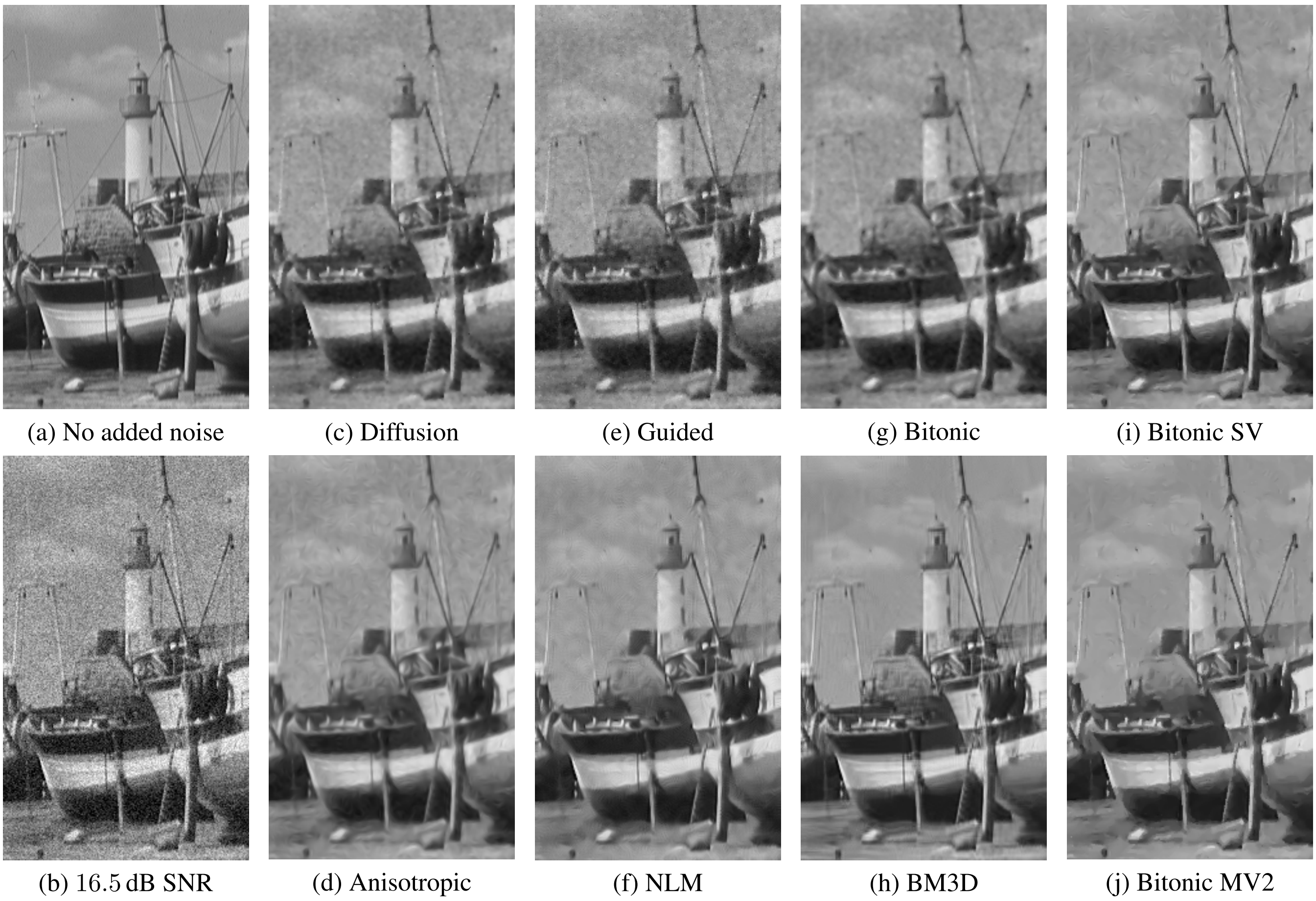
[6] https://homepages.cae.wisc.edu/~ece533/images/ and http://decsai.ugr.es/cvg/CG/base.htm

Table 1: Signal to noise ratio (SNR), Structural Similarity (SSIM) and processing times for images with added Gaussian noise σ. The best three algorithms in each case are highlighted in bold, with the rank given by the superscript.

Noise σ		0.01			0.04			0.16			0.64	
	SNR	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**
	dB		secs	dB		secs	dB		secs	dB		secs
boat (512 x 512, grey)												
Original	34.63	0.965	-	22.57	0.682	-	10.55	0.199	-	-0.42	0.025	-
BM3D	**35.82**^{1}	**0.976**^{1}	1.75	**28.38**^{1}	**0.890**^{1}	1.92	**22.15**^{1}	**0.730**^{1}	2.53	**16.12**^{1}	**0.499**^{2}	2.58
NLM	34.63	0.965	0.12	**27.55**^{2}	**0.867**^{2}	1.21	20.69	0.631	2.51	15.00	0.447	66.33
Diffusion	**35.46**^{2}	**0.972**^{2}	0.48	27.01	0.859	0.33	20.39	0.638	0.11	15.71	0.484	0.60
Guided	**35.10**^{3}	0.968	0.06	26.91	0.857	0.04	19.19	0.526	0.04	12.47	0.172	0.08
Anisotropic	34.47	0.965	0.09	26.37	0.852	0.10	20.87	0.669	0.21	15.66	**0.492**^{3}	0.68
Bitonic	34.47	0.965	0.17	25.97	0.825	0.16	19.98	0.638	0.18	15.63	0.482	0.23
Bitonic T	34.53	0.968	0.32	26.60	0.851	0.22	20.12	0.639	0.26	15.69	0.480	0.28
Bitonic SV	34.47	0.965	0.46	26.81	0.854	0.76	21.24	0.683	2.14	15.72	0.482	5.23
Bitonic SVT	34.94	**0.970**^{3}	2.31	27.13	**0.864**^{3}	1.29	21.27	0.683	2.36	15.72	0.482	5.32
Bitonic MV1	34.94	**0.970**^{3}	2.31	**27.13**^{3}	0.864	1.58	**21.37**^{3}	**0.685**^{3}	1.70	**15.97**^{3}	0.486	4.82
Bitonic MV2	34.94	**0.970**^{3}	2.31	27.06	0.860	3.04	**21.38**^{2}	**0.691**^{2}	3.60	**16.08**^{2}	**0.500**^{1}	7.87
shapes (128 x 128, grey)												
Original	36.75	0.943	-	24.72	0.601	-	12.75	0.190	-	1.78	0.023	-
BM3D	**46.88**^{1}	**1.000**^{1}	0.08	**39.00**^{1}	**0.995**^{1}	0.09	**27.35**^{1}	**0.912**^{1}	0.13	**17.78**^{1}	**0.615**^{1}	0.13
NLM	44.91	0.996	0.07	34.70	0.976	0.33	22.49	0.751	0.33	16.77	0.543	3.92
Diffusion	41.93	0.984	0.05	34.41	0.974	0.05	22.10	0.732	0.03	**17.53**^{3}	**0.592**^{3}	0.05
Guided	43.22	0.993	0.01	30.88	0.898	0.00	21.42	0.526	0.01	14.49	0.179	0.03
Anisotropic	36.61	0.943	0.03	29.18	0.928	0.03	23.46	0.800	0.04	17.06	**0.608**^{2}	0.07
Bitonic	40.78	0.988	0.03	30.61	0.916	0.02	22.01	0.715	0.03	17.11	0.586	0.03
Bitonic T	46.05	**0.999**^{2}	0.04	33.69	0.977	0.04	22.03	0.715	0.04	17.33	0.572	0.04
Bitonic SV	43.32	0.997	0.10	34.87	0.976	0.11	25.34	0.871	0.26	17.08	0.520	0.34
Bitonic SVT	**46.86**^{2}	0.998	0.34	37.31	0.984	0.26	25.36	0.869	0.30	17.08	0.520	0.32
Bitonic MV1	45.55	0.997	0.31	**38.03**^{3}	**0.984**^{3}	0.22	**25.75**^{3}	**0.885**^{3}	0.41	17.16	0.542	0.40
Bitonic MV2	**46.59**^{3}	**0.998**^{3}	0.68	**38.97**^{2}	**0.990**^{2}	0.46	**26.54**^{2}	**0.908**^{2}	0.43	**17.57**^{2}	0.579	0.42
house (256 x 256, grey)												
Original	35.10	0.948	-	23.11	0.592	-	10.99	0.157	-	0.09	0.021	-
BM3D	**37.95**^{1}	**0.979**^{1}	0.46	**31.57**^{1}	**0.923**^{1}	0.49	**25.60**^{1}	**0.831**^{1}	0.61	**17.56**^{1}	**0.625**^{2}	0.56
NLM	36.11	0.969	0.10	**30.32**^{2}	**0.893**^{2}	0.26	22.69	0.759	2.45	15.67	0.560	15.59
Diffusion	**37.00**^{2}	0.970	0.10	29.01	0.873	0.07	21.49	0.723	0.04	16.68	0.601	0.10
Guided	**36.70**^{3}	0.968	0.01	28.31	0.866	0.01	20.43	0.507	0.01	13.21	0.166	0.01
Anisotropic	34.95	0.946	0.03	28.60	0.868	0.06	22.60	0.765	0.13	16.26	**0.624**^{3}	0.26
Bitonic	34.95	0.946	0.06	28.40	0.855	0.05	21.65	0.716	0.06	16.24	0.611	0.07
Bitonic T	36.15	0.969	0.06	28.45	0.873	0.06	21.44	0.722	0.08	16.66	0.595	0.06
Bitonic SV	35.55	0.960	0.12	29.54	0.878	0.29	23.80	0.797	1.25	16.81	0.599	1.41
Bitonic SVT	36.41	**0.970**^{2}	0.62	29.60	**0.886**^{3}	0.53	23.86	0.793	1.24	16.81	0.599	1.41
Bitonic MV1	36.41	**0.970**^{2}	0.53	**29.61**^{3}	0.885	0.87	**24.05**^{3}	**0.797**^{3}	1.21	**17.35**^{3}	0.621	1.20
Bitonic MV2	36.41	**0.970**^{2}	0.53	29.56	0.885	1.65	**24.40**^{2}	**0.807**^{2}	1.78	**17.43**^{2}	**0.654**^{1}	2.37
varying (168 x 168, grey)												
Original	13.14	0.565	-									
BM3D	**29.44**^{1}	**0.968**^{1}	0.24									
NLM	18.85	0.752	0.02									
Diffusion	18.79	0.734	0.01									
Guided	19.08	0.748	0.00									
Anisotropic	21.46	0.859	0.03									
Bitonic	19.93	0.798	0.04									
Bitonic T	19.93	0.798	0.03									
Bitonic SV	24.05	0.922	0.40									
Bitonic SVT	24.05	0.922	0.42									
Bitonic MV1	**24.64**^{3}	**0.932**^{3}	0.47									
Bitonic MV2	**25.39**^{2}	**0.948**^{2}	0.58									

Table 2: Signal to noise ratio (SNR), Structural Similarity (SSIM) and processing times for images with added Gaussian noise σ. The best three algorithms in each case are highlighted in bold, with the rank given by the superscript.

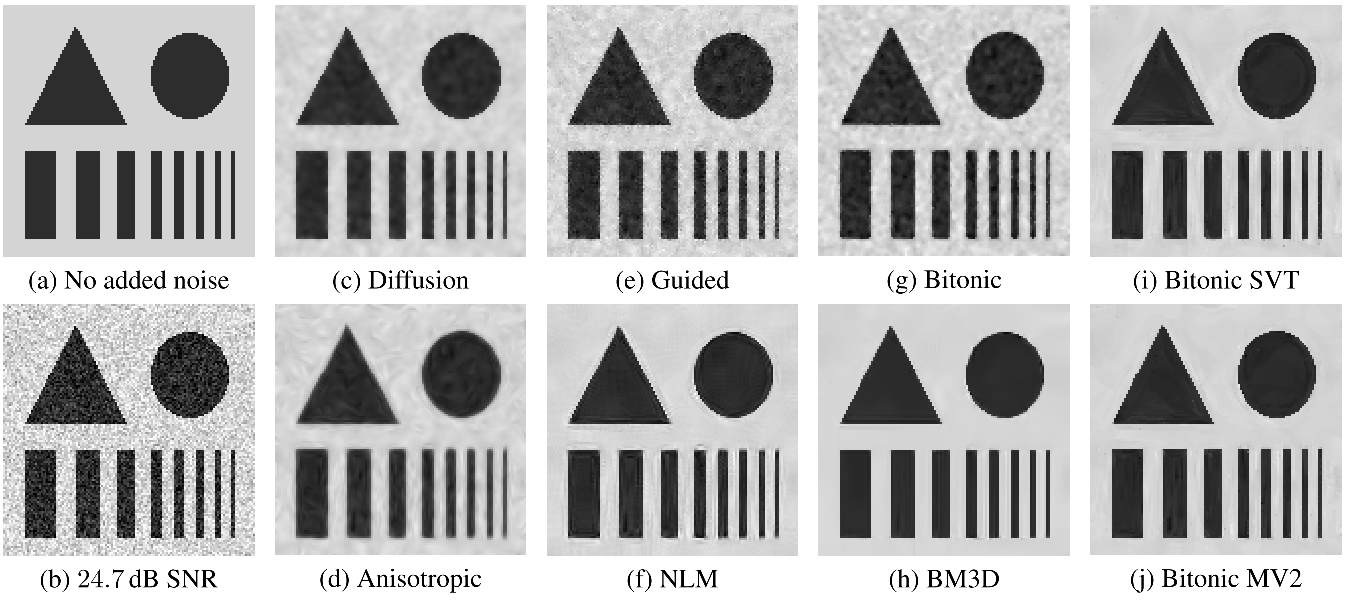
Noise σ		0.01			0.04			0.16			0.64	
	SNR	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**	**SNR**	**SSIM**	**Time**
	dB		secs	dB		secs	dB		secs	dB		secs
fruits (512 x 512, colour)												
Original	36.09	0.982	-	24.45	0.793	-	12.47	0.262	-	1.40	0.043	-
BM3D	**38.63**^{1}	**0.987**^{1}	3.57	**32.11**^{1}	**0.936**^{1}	3.31	**26.66**^{1}	**0.840**^{1}	6.57	**18.55**^{3}	0.685	4.37
NLM	36.67	0.980	0.20	**30.58**^{2}	**0.924**^{2}	2.64	25.23	0.806	14.44	17.55	0.651	192.48
Diffusion	**37.69**^{2}	**0.986**^{2}	2.51	29.98	0.923	0.99	23.27	0.774	0.50	17.64	0.671	2.48
Guided	37.44	0.981	0.12	29.83	0.920	0.09	21.93	0.697	0.09	14.62	0.297	0.14
Anisotropic	36.29	0.982	0.17	29.48	0.919	0.24	24.26	0.812	0.66	18.17	0.684	1.70
Bitonic	36.29	0.982	0.47	29.14	0.908	0.46	22.94	0.775	0.51	17.69	0.671	0.70
Bitonic T	37.25	0.984	0.67	29.82	0.919	0.63	23.30	0.778	0.66	17.72	0.670	0.64
Bitonic SV	36.29	0.982	1.17	30.25	0.920	2.14	25.17	0.822	4.80	18.49	**0.687**^{3}	8.51
Bitonic SVT	**37.46**^{3}	**0.986**^{3}	3.22	30.51	**0.924**^{3}	3.19	25.24	0.823	5.06	18.49	**0.687**^{3}	8.53
Bitonic MV1	**37.46**^{3}	**0.986**^{3}	3.17	**30.55**^{3}	0.923	3.83	**25.42**^{3}	**0.824**^{3}	5.34	**18.90**^{2}	**0.691**^{2}	8.08
Bitonic MV2	**37.46**^{3}	**0.986**^{3}	3.15	30.38	0.924	5.39	**25.51**^{2}	**0.828**^{2}	8.14	**19.02**^{1}	**0.700**^{1}	13.95
monarch (512 x 768, colour)												
Original	33.01	0.981	-	21.16	0.784	-	9.23	0.297	-	-1.79	0.062	-
BM3D	**37.47**^{1}	**0.988**^{2}	4.61	**31.09**^{1}	**0.969**^{1}	4.98	**23.93**^{1}	**0.921**^{1}	9.92	**15.21**^{1}	**0.757**^{1}	6.66
NLM	35.13	0.986	1.26	27.85	**0.963**^{2}	10.17	**21.89**^{2}	0.890	41.64	13.18	0.661	120.77
Diffusion	**35.33**^{3}	**0.989**^{1}	3.85	27.84	0.956	2.32	19.81	0.839	0.80	13.54	0.710	3.06
Guided	**35.70**^{2}	0.987	0.16	27.25	0.948	0.13	18.26	0.739	0.14	10.93	0.328	0.21
Anisotropic	33.04	0.981	0.30	26.19	0.943	0.39	20.68	0.870	0.77	13.99	0.717	1.62
Bitonic	33.04	0.981	0.70	26.11	0.933	0.70	19.09	0.843	0.73	13.63	0.704	0.95
Bitonic T	34.47	0.987	0.86	27.19	0.953	0.94	19.46	0.847	1.04	13.63	0.704	0.89
Bitonic SV	33.12	0.985	1.40	27.10	0.956	3.59	21.56	0.895	5.76	14.16	0.725	8.86
Bitonic SVT	34.57	**0.987**^{3}	4.21	27.88	0.960	4.86	21.60	0.896	7.31	14.16	0.725	8.88
Bitonic MV1	34.57	**0.987**^{3}	4.11	**27.93**^{3}	0.960	5.12	21.82	**0.896**^{3}	7.70	**14.75**^{2}	**0.729**^{3}	8.68
Bitonic MV2	34.57	**0.987**^{3}	4.13	**28.02**^{2}	**0.961**^{3}	10.69	**21.87**^{3}	**0.900**^{2}	12.02	**14.74**^{3}	**0.747**^{2}	17.21
blue rocks (730 x 1024, colour)												
Original	34.23	0.984	-	22.45	0.817	-	10.54	0.343	-	-0.51	0.060	-
BM3D	**37.24**^{1}	**0.993**^{1}	8.84	**29.22**^{1}	**0.957**^{1}	9.04	**22.76**^{1}	**0.839**^{1}	18.44	**15.81**^{3}	**0.622**^{1}	18.88
NLM	35.22	0.987	7.69	26.93	0.938	7.49	**21.36**^{3}	0.782	41.16	14.76	0.560	369.29
Diffusion	**35.70**^{2}	0.990	7.18	26.69	0.927	2.82	19.84	0.738	1.45	15.05	0.588	7.25
Guided	**35.62**^{3}	0.988	0.34	26.56	0.925	0.23	19.05	0.691	0.24	12.22	0.310	0.50
Anisotropic	34.31	0.984	0.81	25.70	0.912	0.94	20.88	0.783	2.63	15.31	0.600	4.48
Bitonic	34.31	0.984	1.74	24.86	0.893	2.12	19.59	0.731	2.62	15.14	0.587	2.36
Bitonic T	35.10	0.990	2.82	26.21	0.922	2.71	19.75	0.737	2.83	15.14	0.587	2.53
Bitonic SV	34.31	0.984	3.63	26.25	0.921	4.64	21.16	0.798	11.10	15.40	0.600	22.20
Bitonic SVT	35.32	**0.990**^{2}	16.61	27.17	**0.940**^{2}	8.81	21.19	0.800	11.97	15.40	0.600	22.47
Bitonic MV1	35.32	**0.990**^{2}	16.55	**27.20**^{2}	**0.940**^{3}	11.20	21.32	**0.803**^{3}	12.54	**15.88**^{2}	**0.610**^{3}	21.02
Bitonic MV2	35.32	**0.990**^{2}	16.46	**27.18**^{3}	0.939	22.23	**21.43**^{2}	**0.806**^{2}	20.64	**16.02**^{1}	**0.622**^{2}	35.70
south sound (680 x 1024, colour)												
Original	34.52	0.974	-	22.80	0.721	-	10.78	0.180	-	-0.30	0.023	-
BM3D	**40.63**^{1}	**0.990**^{1}	9.81	**34.87**^{1}	**0.968**^{1}	10.93	**29.07**^{1}	**0.937**^{1}	18.51	**19.32**^{3}	0.832	12.17
NLM	**38.34**^{3}	0.985	7.09	**32.84**^{2}	**0.962**^{2}	74.91	26.81	0.912	132.89	18.87	0.816	538.52
Diffusion	37.81	**0.988**^{2}	6.87	31.69	0.958	5.16	24.68	0.895	3.55	18.67	0.829	7.36
Guided	**38.51**^{2}	0.986	0.25	30.63	0.948	0.23	22.07	0.728	0.27	13.88	0.281	0.29
Anisotropic	36.76	0.985	0.39	31.13	0.954	0.97	25.90	0.910	2.48	19.12	**0.849**^{2}	6.46
Bitonic	36.61	0.983	1.56	30.55	0.951	1.98	24.81	0.895	2.11	19.04	0.845	3.17
Bitonic T	37.67	**0.987**^{3}	2.54	31.04	0.953	2.51	24.87	0.892	2.71	18.81	0.832	2.28
Bitonic SV	36.90	0.983	4.15	31.90	0.958	6.68	26.63	**0.918**^{3}	17.85	19.26	0.846	28.77
Bitonic SVT	37.53	0.986	8.17	31.94	0.958	11.56	26.58	0.916	18.84	19.26	0.846	28.75
Bitonic MV1	37.53	0.986	8.05	31.98	0.958	11.66	**27.04**^{3}	0.917	17.25	**19.84**^{2}	**0.848**^{3}	30.88
Bitonic MV2	37.53	0.986	8.09	**32.13**^{3}	**0.959**^{3}	19.25	**27.12**^{2}	**0.920**^{2}	34.15	**19.98**^{1}	**0.855**^{1}	44.74



(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SV

(b) 16.5 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

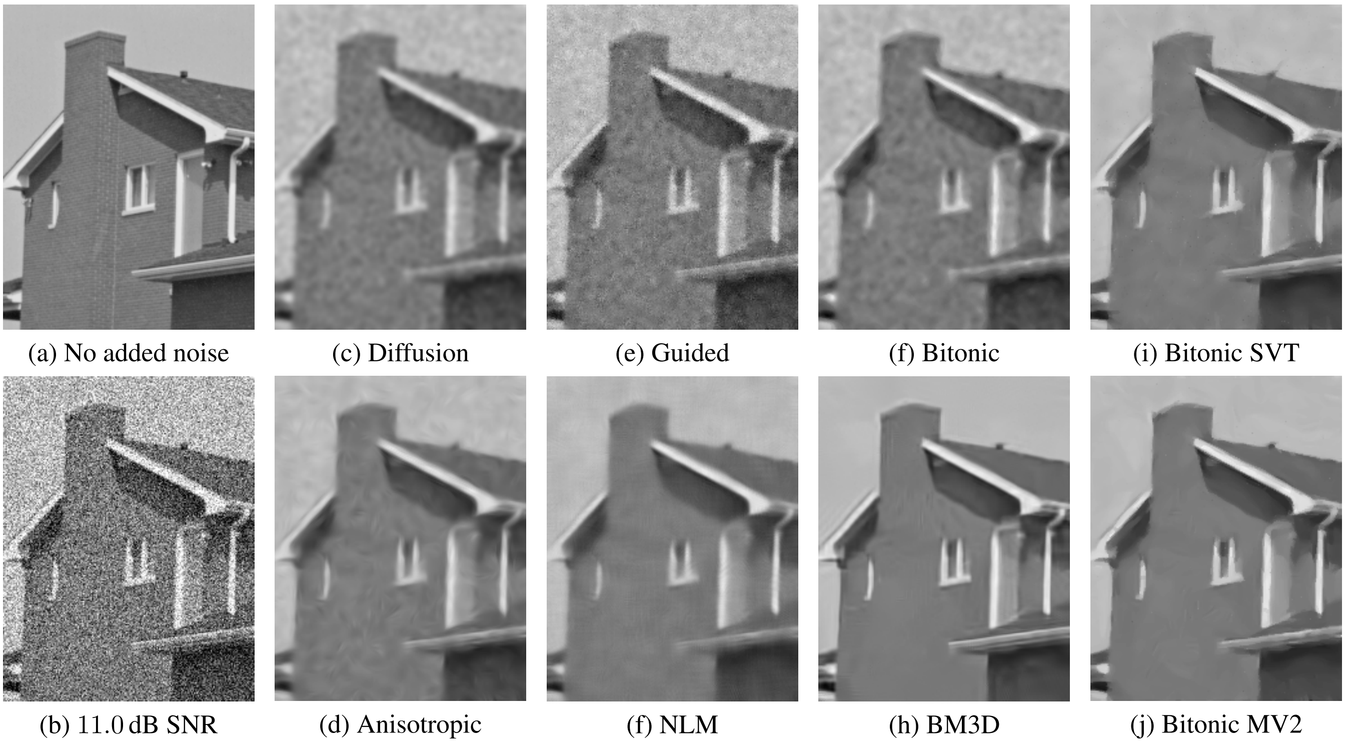
Figure 9: Results for noise in (b) added to section of 'boat' image in (a): see Table 1 for further details.



(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SVT

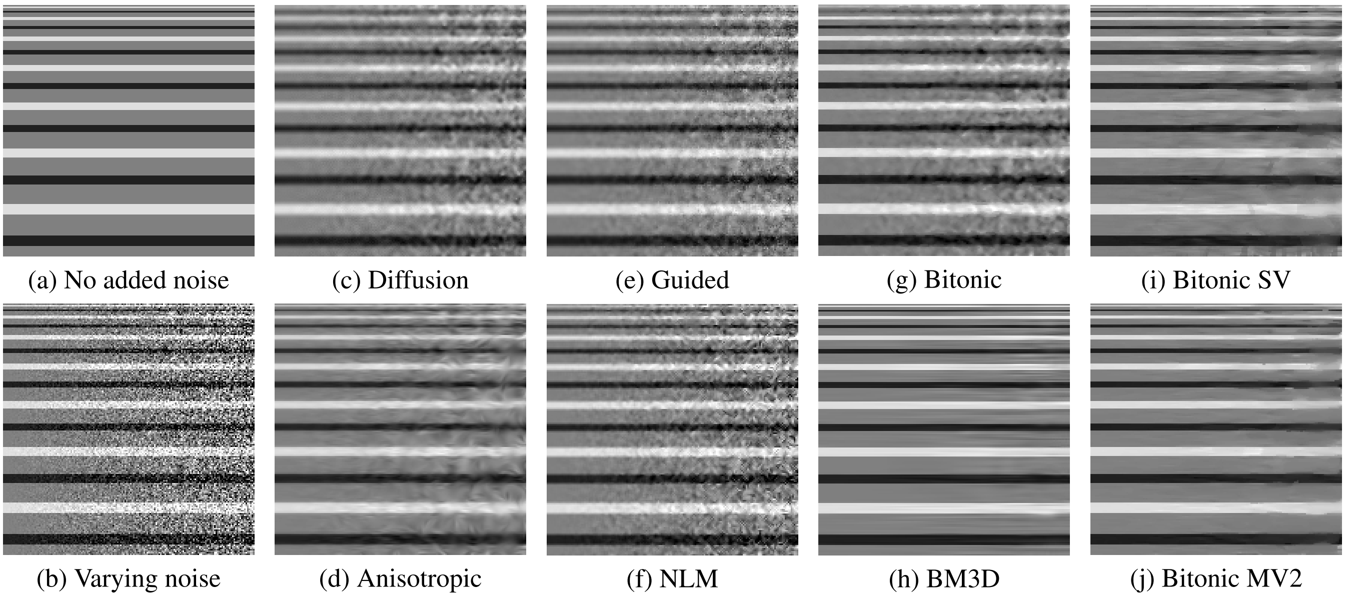
(b) 24.7 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 10: Results for noise in (b) added to 'shapes' image in (a): see Table 1 for further details.


(a) No added noise (c) Diffusion (e) Guided (f) Bitonic (i) Bitonic SVT

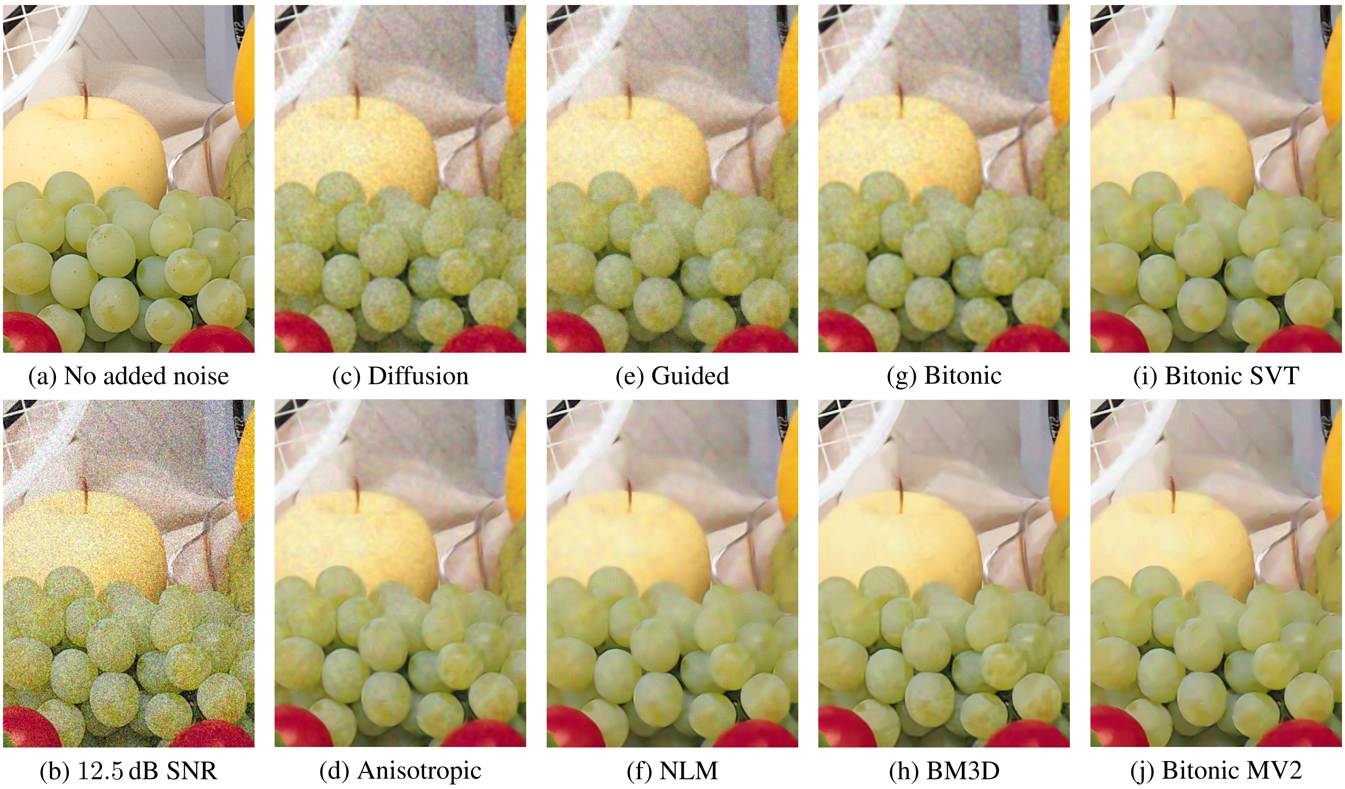
(b) 11.0 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 11: Results for noise in (b) added to section of ‘house’ image in (a): see Table 1 for further details.


(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SV

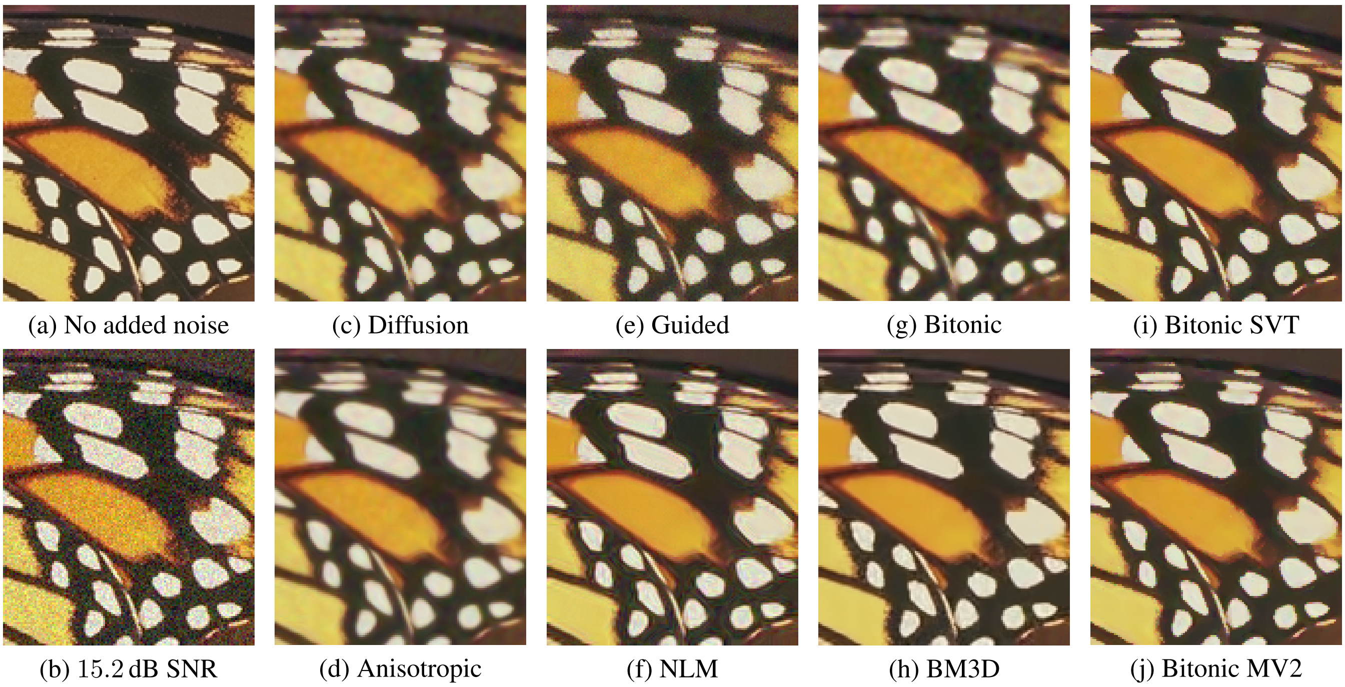
(b) Varying noise (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 12: Results for noise in (b) added to ‘varying’ image in (a): see Table 2 for further details.


(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SVT

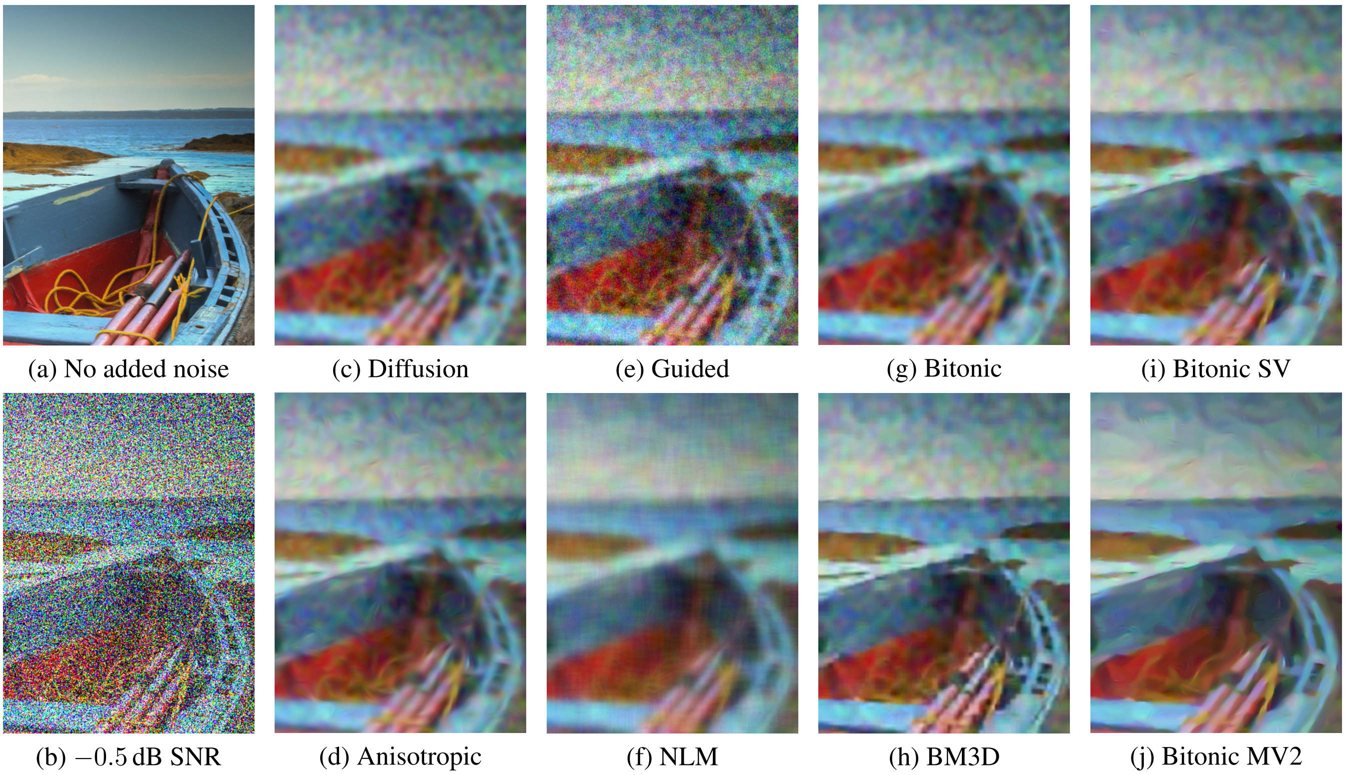
(b) 12.5 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 13: Results for noise in (b) added to section of ‘fruits’ image in (a): see Table 1 for further details.


(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SVT

(b) 15.2 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

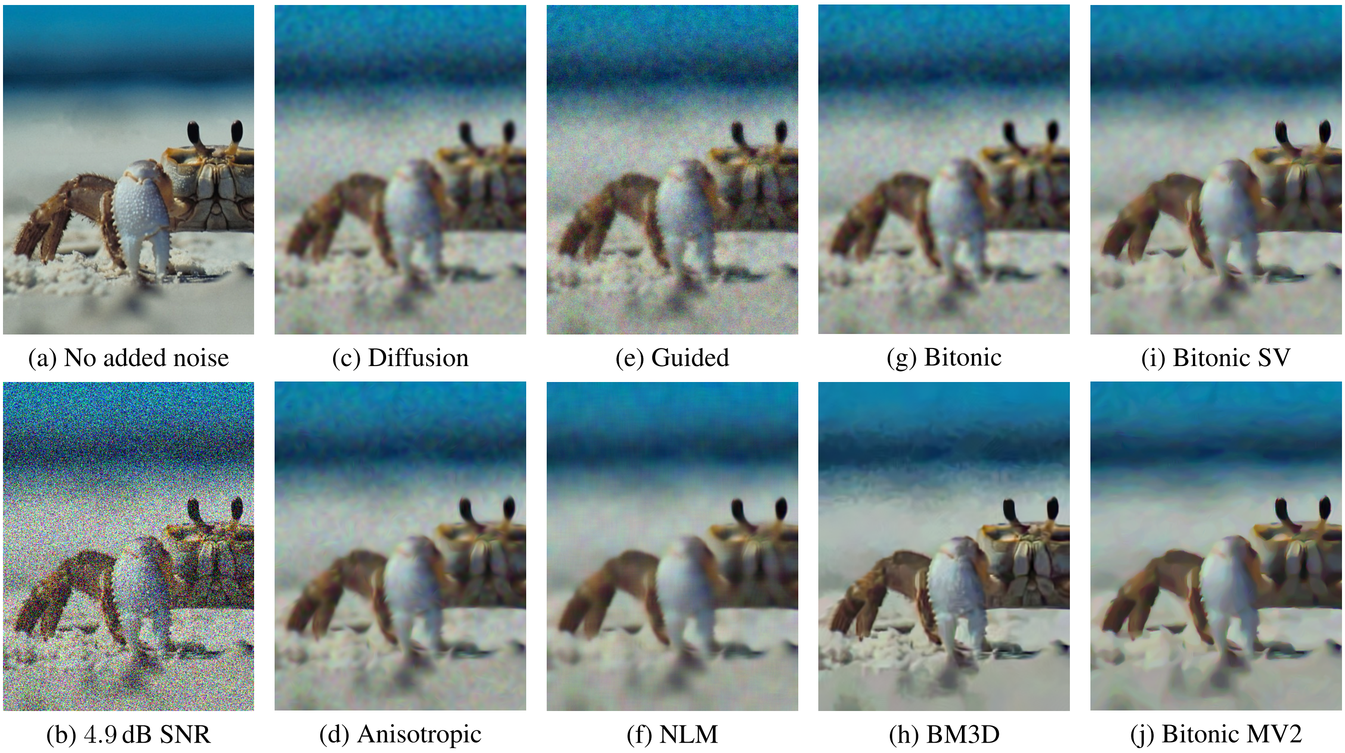
Figure 14: Results for noise in (b) added to section of ‘monarch’ image in (a): see Table 2 for further details.



(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SV

(b) -0.5 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 15: Results for noise in (b) added to section of ‘blue rocks’ image in (a): see Table 2 for further details.



(a) No added noise (c) Diffusion (e) Guided (g) Bitonic (i) Bitonic SV

(b) 4.9 dB SNR (d) Anisotropic (f) NLM (h) BM3D (j) Bitonic MV2

Figure 16: Results for noise in (b) added to section of ‘south sound’ image in (a): see Table 2 for further details.

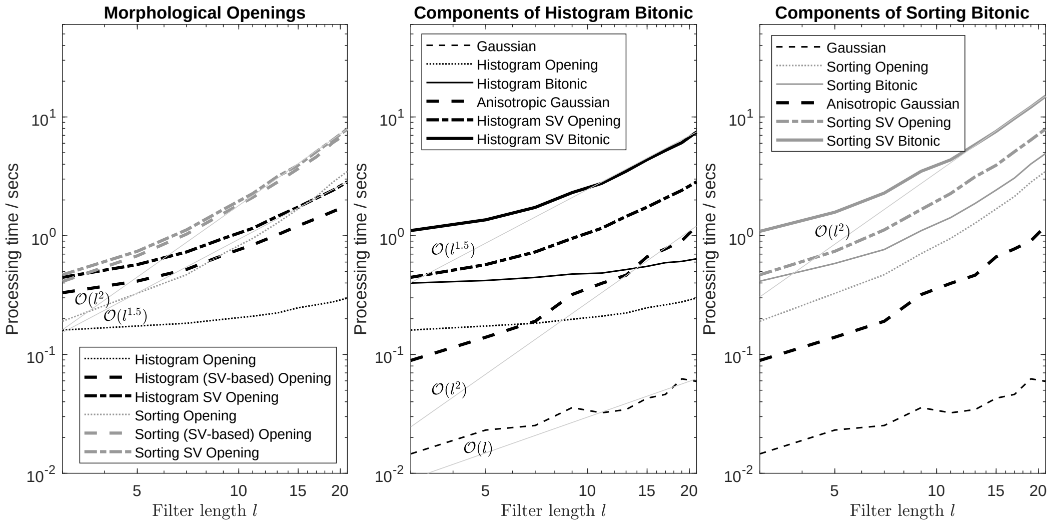


Figure 17: Processing times for the SV bitonic filter. These are for a single 512×512 colour image, for both histogram-based and sorting-based implementations, based on the average of ten results. On the left is the morphological opening which forms the basis of the SV bitonic filter. The dashed lines show a fixed (single mask) filter, but with the SV framework: there is a considerable overhead in storing region indices and checking for mask locations. However the subsequent trialling of up to 32 masks (dash-dot lines) is much less significant. The middle graph shows the components of the bitonic and SV bitonic: whilst the anisotropic Gaussian is much slower than the Gaussian, it is still the opening and closing operations that dominate the time taken for both the bitonic filters. On the right, sorting-based versions of the bitonic are slower, tending to $\mathcal{O}(l^2)$ rather than $\approx \mathcal{O}(l^{1.5})$, in the mask width l.

a slight improvement over Bitonic SVT at most noise levels, for minimal increase in processing time: in fact Bitonic MV1 often requires a smaller mask size than Bitonic SVT, in which case the processing time can even be reduced.

The numerical results are not an adequate summary of the performance, since the characteristic residual noise from each algorithm is very different, even for similar SNR or SSIM values. These characteristics can only be seen in images: they are very visible on the house in Fig. 11. At these noise levels, Diffusion blurs the signal as well as reducing noise, and the data threshold in Guided either leads to much noise surviving (as shown) or too much blurring. Bitonic has sharper edges than Diffusion but Bitonic SVT is a very clear improvement. NLM starts to reveal a cross-hatching which results from finding false patterns in the noise, and BM3D also reveals some false patterning across the front face of the house. Bitonic MV2 has less characteristic noise, though the mask shape (particularly thin ellipses) is evident in some places; however there is also slightly less detail than with BM3D. At a slightly higher noise level, the south sound image in Fig. 16 also demonstrates the contrast to BM3D. Details of the crab claw and underside are preserved since the patterns are sufficiently repeated: Bitonic MV2 shows less detail, but the edges are very clear and there is little characteristic noise across the smooth background. This is a direct result of enforcing bitonicity across a larger scale due to the increased noise, but similar features are still visible on close inspection of the grapes in Fig. 13 for the medium noise case.

The very highest noise levels, as in the blue rocks image in Fig. 15, reveal the block-based structure on which BM3D is based, apparent at the oars and along the edge of the boat. At this level, NLM has poor performance, and prohibitively long processing times. Bitonic MV2, whilst not preserving very small features, otherwise performs very well, recovering edges smoothly with limited characteristic noise across more constant regions. The image in Fig. 12 confirms that Bitonic SV, like Bitonic, is still very good at handling varying noise levels. In this case BM3D also performs well, since the structure is correlated in the horizontal direction, though horizontal streaks are evident as a result of this.

The Bitonic SV has required several novel developments with potentially broader applications. These include an efficient implementation of robust SV opening and closing, with a framework that allows a considerable number of masks to be tested. Together with the careful analysis of relative centiles for different shapes, this has enabled mask choice to be determined from morphological operations, rather than from a smoothed structure tensor. The local anisotropy derived from such a tensor has also been improved at corner locations. Good noise-reduction results have been achieved by using this robust SV with simple elliptical masks, but the framework also opens up the possibility of investigating more complex mask shapes, which may be what is needed to preserve smaller details in the images.

A new formulation has also been presented for non-iterative directional Gaussian smoothing, which can optionally also be thresholded at no additional cost to processing times. Whilst this technique alone does not perform as well as when embedded in Bitonic SV, it still has quite reasonable noise-reduction capability at high noise levels, and is faster than all but Guided, which is not appropriate in these cases.

5 Conclusion

The structurally varying bitonic is a considerable improvement on the fixed bitonic for image noise reduction, whilst increasing the processing time by usually less than an order of magnitude. It is increasingly competitive with linear noise-reduction algorithms, outperforming non-local means at all noise levels and the block-matching 3D filter for very high noise. In such scenarios, the structurally varying bitonic has less characteristic residual noise and very good edge definition, with similar processing times, though with some loss of small scale detail compared to block-matching. The non-iterative directional Gaussian used in the new bitonic filter is much faster and yet also has good performance in high noise. The novel implementation of robust structurally varying opening and closing operations enables future development of the bitonic, and morphological operations more generally, with more complex mask shapes. Inclusion of data thresholding within these opening and closing operations has improved performance in low noise and enables the adoption of a multi-resolution framework around morphological operations for high noise.

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

References

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

Baek, J., Jacobs, D. E., 2010. Accelerating spatially varying gaussian filters. ACM Transactions on Graphics (TOG) 29 (6), 169.

Baghaie, A., Yu, Z., 2015. Structure tensor based image interpolation method. AEU-international Journal of Electronics and Communications 69 (2), 515–522.

Bouaynaya, N., Schonfeld, D., 2008. Theoretical foundations of spatially-variant mathematical morphology part ii: Gray-level images. IEEE Transactions on pattern analysis and machine intelligence 30 (5), 837–850.

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

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.

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.

Ćurić, V., Landström, A., Thurley, M. J., Hendriks, C. L. L., 2014. Adaptive mathematical morphology–a survey of the field. Pattern Recognition Letters 47, 18–28.

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.

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.

Lampert, C. H., Wirjadi, O., 2006. An optimal nonorthogonal separation of the anisotropic gaussian convolution filter. IEEE Transactions on Image Processing 15 (11), 3501–3513.

Landström, A., Thurley, M. J., 2013. Adaptive morphology using tensor-based elliptical structuring elements. Pattern Recognition Letters 34 (12), 1416–1422.

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, 860–872.

Paris, S., Hasinoff, S. W., Kautz, J., 2011. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. ACM Trans. Graph. 30 (4), 68–1.

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.

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

Soille, P., Talbot, H., 2001. Directional morphological filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11), 1313–1329.

Tankyevych, O., Talbot, H., Dokládal, P., Passat, N., 2009. Spatially-variant morpho-hessian filter: Efficient implementation and application. In: International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing. Springer, pp. 137–148.

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.

[7] MATLAB file exchange: Structurally Varying Bitonic Filter by Graham Treece, 16 Aug 2018

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

Treece, G. M., Feb. 2017. Refinement of clinical x-ray computed tomography (CT) scans containing metal implants. Computerized Medical Imaging and Graphics 56, 11–23.

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

Verd-Monedero, R., Angulo, J., Serra, J., 2011. Anisotropic morphological filters with spatially-variant structuring elements based on image-dependent gradient fields. IEEE Transactions on Image Processing 20 (1), 200–212.

Wang, Z., Bovik, A. C., Sheikh, H. R., Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing 13 (4), 600–612.

Weickert, J., 1998. Anisotropic diffusion in image processing. Vol. 1. Teubner Stuttgart.

Wong, W. C., Chung, A. C., Yu, S. C., 2004. Trilateral filtering for biomedical images. In: Biomedical Imaging: Nano to Macro, 2004. IEEE International Symposium on. pp. 820–823.