Image denoising with Weighted ORientation-Matched filters(WORM)

Debajyoti Karmaker*, Ingo Schiffner‡, Michael Wilson*and Mandyam V. Srinivasan*†
* The Queensland Brain Institute, University of Queensland,St Lucia, QLD, Australia
† The School of Information Technology and Electrical Engineering, University of Queensland, Australia
‡School of Natural Sciences, Bangor University, Gwynedd, Wales, UK
Email: {d.karmaker,m.wilson,m.srinivasan}@uq.edu.au and i.schiffner@bangor.ac.uk

***Abstract*—Real world signals commonly exhibit slow variations or oscillations, punctuated with rapid transients. For example, images typically have smooth regions interrupted by edges or abrupt changes in contrast. These abrupt changes are often the most interesting parts of the data perceptually, as well as in terms of the information that they provide. Some of the high frequency content represents the important abrupt changes in image intensity that are associated with real edges of objects in the image. However, some of the high-frequency content also comprises the noise that is present in the image. We wish to retain this edge information, while removing the noise. In this paper, we present a dynamic filtering process where the dynamic mask is oriented to match the local gradients and its weights are proportional to the magnitude of the local gradients.**

Keywords*—Image denoising, Non-local filters, Spatial filtering, Noise removal, Edge detection.

I. Introduction

The use of Unmanned Aerial Vehicles (UAVs) is becoming more and more pervasive every year, rendering current ground-based control systems inadequate in avoiding mid-air collisions. Most airborne creatures, such as birds or flying insects, are extremely adept at avoiding collisions with their conspecifics or other moving objects in their environments. How they achieve this is largely unknown. In the past decade, researchers have started to gain insights into how birds and insects control their flight speed [1], avoid obstacles [2] and perform smooth landings [3]. This has drawn considerable attention from roboticists, who are challenged with similar problems in the design of guidance systems for unmanned aerial vehicles. However, studying bird flight is not simple: one first has to be able to collect accurate data from flying animals, with sufficient temporal and spatial resolution. This is best achieved using high-speed cameras and stereo or multi-camera setups. While high-speed cameras are now becoming more and more affordable, the need for efficient image restoration methods has grown with the use of high-speed video cameras as the exposure time for high-speed cameras is limited by the frame rate, which limits the SNR. Thus, pre-processing the images to get denoised version of the images is often the first step conducted before the images data is analysed. Natural images mostly contain additive random noise which can be modelled as a Gaussian. Speckle noise [4] is observed in ultrasound images whereas Rician noise [5] affects MRI images. The scope of the paper is to focus on a smoothing technique to denoise the images acquired by high speed cameras to help detecting edges, often filmed in poor conditions.

II. Related Works

There is substantial amount of work available on various image denoising techniques. Denoising techniques can be broadly categorized into two approaches: (A) spatial filtering methods and (B) transform domain filtering methods.

A. Spatial Filtering

Spatial filters are widely used till days before edge detection algorithms are applied. These methods have less computational complexity which is most suitable as a pre-processing technique. These methods remove noise by convolving the original image with a mask (sliding window). Spatial filters can be further sub divided into two categories: (1) Linear filters and (2) non-linear filters.

1) Linear Filters: One form of linear filtering is the average filter or mean filter. A mean filter acts on an image by smoothing it; that is, it reduces the intensity variation between adjacent pixels. The mean filter is nothing but a simple sliding window spatial filter that replaces the center value in the window with the average of all the neighbouring pixel values including itself.

Arguably the most widely used linear filter applied before edge detection is Gaussian smoothing [6], a 2D convolution operator - used to smooth images and reduce the noise contained in an image. The technique is very much similar to the mean filter, but it uses a special kernel that represents the shape of a Gaussian (bell-shaped). Gaussian smoothing makes use of the 2D distribution as a point-spread function, and this is accomplished by convolution.

Another variation of linear filters are the adaptive filters. Adaptive filters are adept at denoising images with abrupt changes in intensity. This kind of filter can handle irregularity in a signal with little a prior knowledge about the signal to be processed [7]. The Least Mean Square (LMS) adaptive filter works well for images corrupted with salt and pepper type noise and it does a better denoising job compared to the mean filter.

The most important advantages of these above-mentioned filters are their high speed and their limited computational complexity. An accompanying disadvantage is that linear filters tend to blur sharp edges, destroy lines and other fine image details, and perform poorly in the presence of signal-dependent noise.

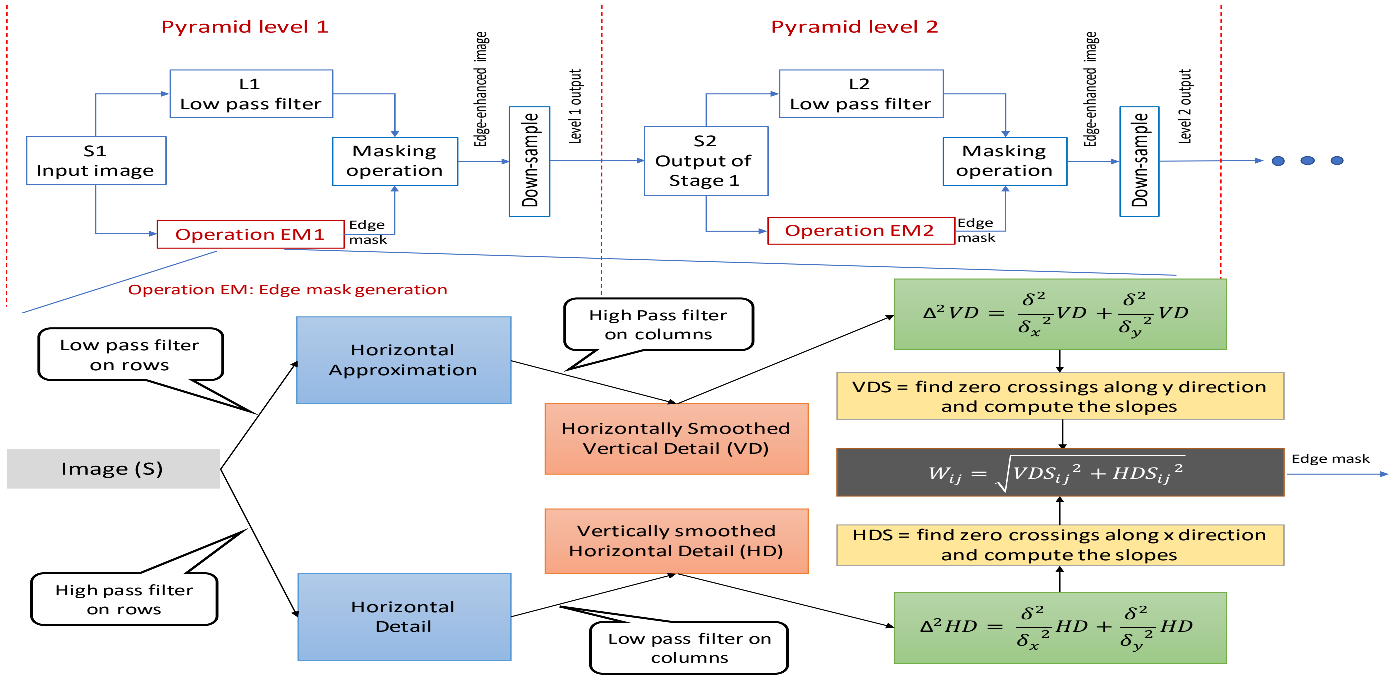


Fig. 1: The figure contains an system overview (top diagram) and detail (lower diagram)

2) Non-linear Filters: Largely linear filters eliminate noise to a reasonable level but they achieve this at the expense of blurring images. A variety of nonlinear median- type filters such as weighted median [8], rank conditioned rank selection [9], and relaxed median [10] have been proposed to handle the limitations of linear filters.

The median filter also engages the mask approach similar to the mean filter. The center pixel under the mask is replaced with the median value of the pixel values that belong inside the mask. As the median value is not significantly influenced by an outlier in a neighbourhood, the median filter is more robust compared to the mean filter. For the same reason, the median filter also performs significantly better while preserving the sharp edges.

The spatial median filter is another variation of non-linear filters. In this filter, the median value is computed by computing the spatial depth between a point and a set of points in a neighbourhood. The central pixel inside a mask is judged to be corrupted or not based on these spatial depth values. The central pixel will remain unchanged if the pixel is not corrupted.

There are various implementations of weighted median filters (WMF) available. The weighted median filters give more weight to some values within the window. The centre weighted median filter is an extension of the weighted median filter where weight is given to the central value of a window and is thus easier to design and implement than other versions of weighted median filters.

B. Transform Domain Filtering

Amongst various methods denoising under transform domain filtering, the most popular is wavelet transform [11]. The principle idea behind wavelet transform is to break up a signal into different frequency components. Next, each section is analysed with a resolution that matches its scale. The effectiveness of this method lies in its capability of representing the signal in few transform coefficient values. Wavelets provide some advantages over Fourier transforms. For example, they do a good job in approximating signals with sharp spikes or signals with discontinuities. The wavelet equation produces different wavelet families like Daubechies, Haar, Coiflets, etc. [12]. But these methods have high run time complexity and also depend on the cut-off frequency and the filter function behaviour. Furthermore, they may produce artificial frequencies in the processed image.

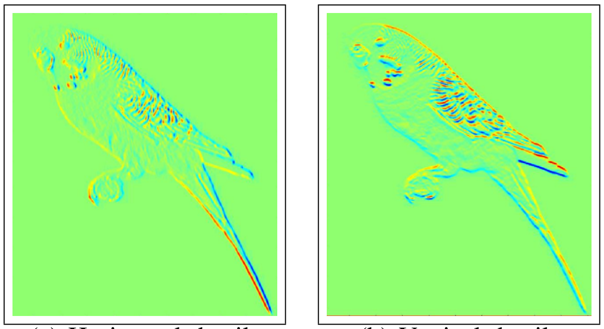
III. Image Denoising using Weighted Orientation-Matched Filters (WORM)

Figure 1 describes the pyramidal system architecture of the proposed image denoising technique. Given an image S, the image is first filtered with a low-pass filter, which removes the high spatial frequencies whilst preserving the low-frequency components of the image. Concurrently, an edge mask W (see Figure 1, Operation EM) is generated from the input image S. To create the edge mask, we apply a low pass filter and a high pass filter to the rows of the image S. The low pass filter extracts the low-frequency components (Horizontal Approximation) and the high pass filter extracts the high-frequency components (Horizontal Detail). We then apply a high pass filter to the columns of the Horizontal Approximation, which yields a horizontally smoothed Vertical Detail (VD) of the original Horizontal Approximation. Similarly, by applying a low pass filter to the columns of the Horizontal Detail, we obtain a vertically smoothed Horizontal Detail (HD). In effect, this is a rapid way of low-and high-pass filtering the image S in two dimensions that speeds up computation by taking

$$P_L(x,y) = \sum_{m=-2}^{2} \sum_{n=-2}^{2} \left(\frac{W_{L-1}(2x+m, 2y+n) + W_{L-1}(2x-m, 2y-n)}{2} \right) LPFO_{L-1}(2x+m, 2y+n) \quad (1)$$

$$P_L(x) = \sum_{m=-2}^{2} \left(\frac{W_{L-1}(2x+m) + W_{L-1}(2x-m)}{2} \right) LPFO_{L-1}(2x+m) \quad (2)$$

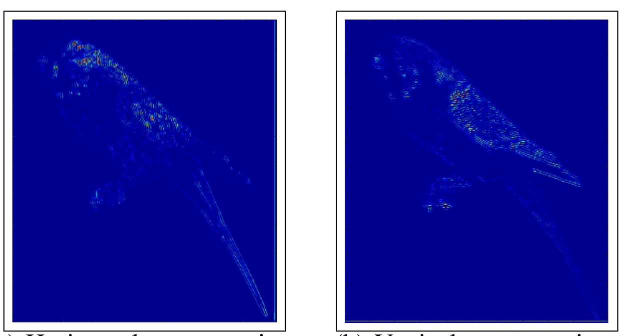
advantage of the separability of the kernels of the low and high-pass filters. The results are shown in Figure 2.



(a) Horizontal details (b) Vertical details

Fig. 2: Horizontal and Vertical details

Next, the Laplacians $\Delta^2 VD$ and $\Delta^2 HD$ are computed from the VD and HD images. We then construct Vertical Detail Slopes (VDS) by finding the gradients of the VD image at the zero crossings of $\Delta^2 VD$ along the y axis, Similarly, we build the Horizontal Detail Slopes (HDS) by scanning the zero crossings of the HD image along the x axis. Figure 3, shows the detected zero crossings, labeled according to the magnitudes of the slopes. The slopes are represented in a heat map where red represents the strongest slopes.



(a) Horizontal zero crossings (b) Vertical zero crossings

Fig. 3: Horizontal and Vertical zero crossings

Finally, the edge mask (W) is created by taking the magnitudes of the slope values from HDS and VDS using equation $W_{ij} = \sqrt{HDS_{ij}^2 + VDS_{ij}^2}$ which represents the magnitude of the local two-dimensional gradient of the image. Figure 4 shows the weight values of the mask, again plotted as a heat map.

Once the edge mask (W) is generated, it is convolved with

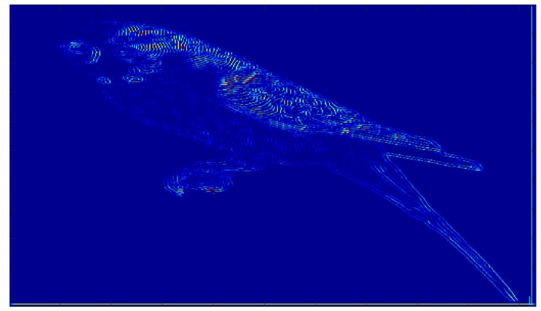


Fig. 4: Edge mask

the low-pass filtered output (LPFO) and downsampled by a factor of two to produce the input for the next pyramid level. Unlike normal convolution - where the spatial filter is constant irrespective of image location, here, the filter is dynamically generated from the edge mask corresponding to each image location. At each image location the filter is oriented to match the local image gradient, and its weight(gain) is proportional to the magnitude of the local gradient. It is a Weighted, ORientation-Matched filter, which we acronymize as WORM. The edge-masked image is then down-sampled by a factor of two in the horizontal and vertical dimensions, to yield an edge-masked image that contains 1/4 as many pixels as the original image.

The processes of dynamic convolution and downsampling is represented by expressions 1. where P_L refers to the input image for the L^{th} level of the pyramid. The $2x$ and $2y$ terms achieve the desired downsampling. The double summations refer to the 5*5 mask that is applied across the rows and columns. For simplicity, we can consider the one-dimensional version of the same expression shown in equation 2.

When $x = 2$, the expression 2 can be written as

$$\begin{aligned} P_L(2) = & \frac{W_{L-1}(2) + W_{L-1}(6)}{2} LPFO_{L-1}(2) \\ & + \frac{W_{L-1}(3) + W_{L-1}(5)}{2} LPFO_{L-1}(3) \\ & + \frac{W_{L-1}(4) + W_{L-1}(4)}{2} LPFO_{L-1}(4) \\ & + \frac{W_{L-1}(5) + W_{L-1}(3)}{2} LPFO_{L-1}(5) \\ & + \frac{W_{L-1}(6) + W_{L-1}(2)}{2} LPFO_{L-1}(6) \end{aligned}$$

In effect, we are reducing the resolution to generate the next level of the image. However, simple downsampling (discarding alternate rows and columns) can create problems, such as aliasing, and loss of useful information. Smoothing (low-pass-filtering) the image prior to downsampling would solve the aliasing problem. This smoothing process is illustrated in Figure 3 for one dimension. The intensity value at pixel 2 in the level L+1 is obtained by calculating a weighted sum of the intensity values at pixels 2,3,4,5 and 6 from the low pass filtered output (LPFO) in level L, as shown in Figure 5. (This process obviously discards the border pixels in the level L+1).

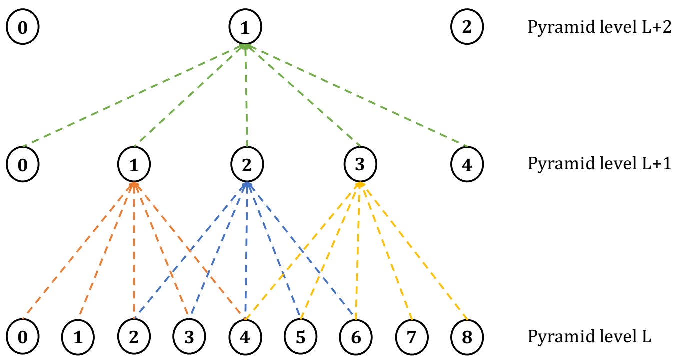


Fig. 5: Pixel intensity contribution from the previous level

A low pass filter of this kind acts to (a) reduce the noise and (b) minimize the effects of aliasing that could arise from the down-sampling process. A unique feature of this filter, however, is that it varies from location to location depending upon the local structure of the image, and allows the filter to selectively enhance local high contrast edges without diminishing their contrast in the way a standard low-pass filter does. This is achieved by using the edge mask to control the weights of the low-pass filter, as shown in Figure 6. The mask weights on either side of the center of the kernel of the low pass filter are made symmetrical, as indicated by the term $\sum_{m=-2}^{2}\left(\frac{W(2x+m)+\ W(2x-m)}{2}\right)$ in equation 2, to ameliorate the effects of noise.

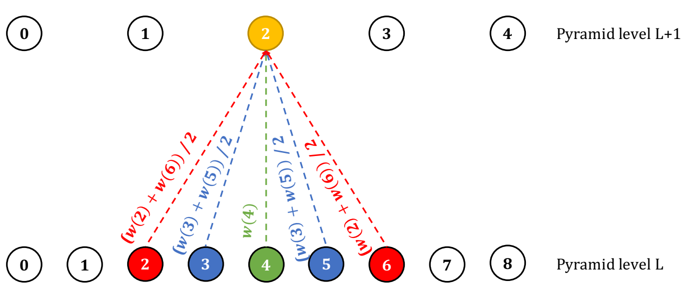
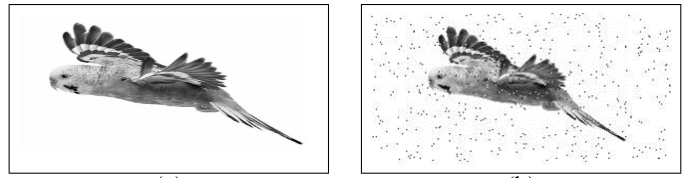


Fig. 6: Dynamic weights assignment

To illustrate the entire process described above, we take an image (Figure 7a) and add some salt and pepper noise using the imnoise function from Matlab. The parameter noise density(d) of imnoise was set to 0.10. This affected approximately 10% pixels of the original image of dimension 256x128 as shown in figure 7b. The noisy image is now passed through a low pass filter and the edge mask (W) is generated from the image as shown in Figure 8. The symbol represents the dynamic convolution and down sampling process described above, which produces an edge-enhanced and down-sampled



(a) (b)

Fig. 7: a) Original image b) Salt & pepper noise added version

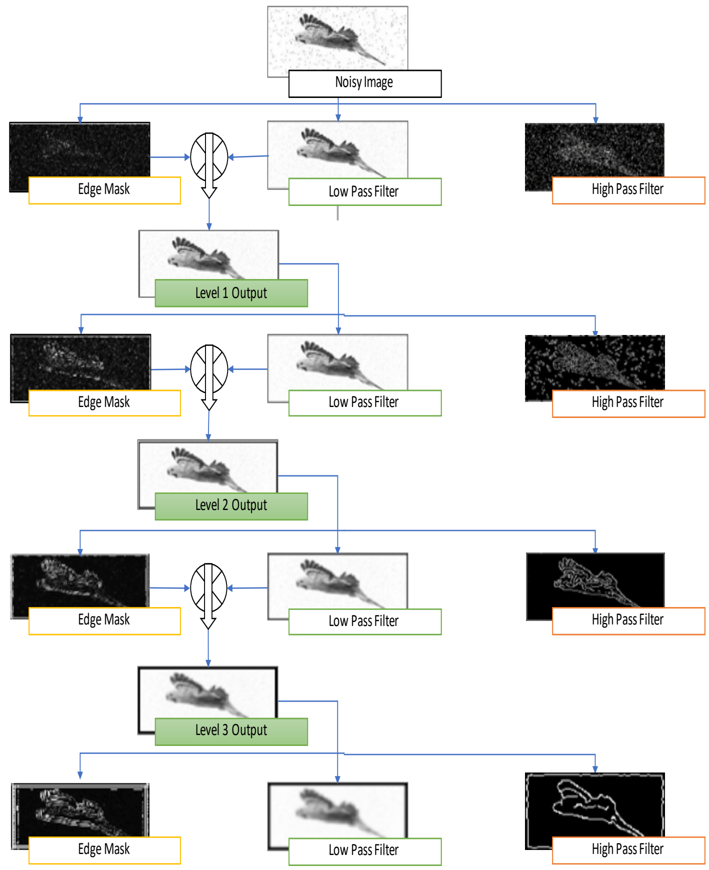


Fig. 8: High pass filter output shows that the noise reduces as the level increases

version of the original image, using the equation described above, for processing at the next level. Additionally, a high pass filter output is also shown at each level in Figure 8 . This high pass filter is required for two reasons. Firstly, the noise reduces significantly as the level increases - as can be seen from the high pass filter outputs (figure 8). Going too far down the pyramid level will eventually get rid of some true edges. For this reason, visual examination of the high pass filter output - allows us to determine the coarsest level of the pyramid that we need to construct. For the particular example shown in Figure 8, level 3 is visually chosen to be the coarsest level, as some of the true edges have already disappeared at level 4 and the input image (unfiltered) at level 3 is nominated as the denoised version of the original noisy image. Secondly, we can apply a threshold on the final high pass filtered output to further reduce the noise and then by combining outputs of the thresholded high pass and low pass filters (at the highest level of the pyramid), we produce a denoised version of the original noisy image.

IV. Experimental Results

To test our smoothing technique, we take one row of an image as shown in Figure 9a and 9c. We add salt and pepper noise by using the imnoise function of Matlab to the image as shown in figure 9b with the parameter noise density(d) set to 0.10 and pick the same row shown in Figure 9d. We compare our method of smoothing and denoising with other popular image smoothing and denoising methods, as shown in Figure 10. The signal to noise ratio is computed by $SNR_{DB} = \frac{-20*log_{10}norm(abs(original-new))}{norm(original)}$.

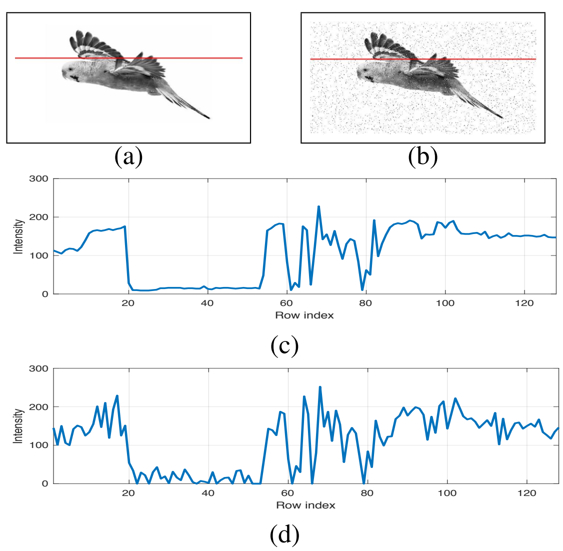


Fig. 9: a) Single row from the original image shown in red b) Same row selected from the Salt & pepper noise added version of the same image c) Single row from the original image where the y axis represents intensity value and the x axis represents row index d) Single row from the noisy image where the y axis represents intensity value and the x axis represents row index

In order to test our denoising method on images captured by high speed cameras, we apply our method to one of our previous works on object detection [13] using interframe differencing. Figure 11 illustrates the benefits of object detection [13] when it is performed on images that have initially been denoised with our denoising method. Here we also show, how our denoising method applied to the same dataset [13] affects detection accuracy. To test robustness of our denoising method, we added two different types of noise (i) salt and pepper noise (SPN)and (ii) Gaussian white noise (GWN) to the images by using the imnoise function of Matlab, with various settings for parameter noise density(d) for SPN and the parameter mean (m) for GWN. Table I evaluates the performance of our denoising method by comparing the detection accuracies with and without denoising, for a range of noise densities.

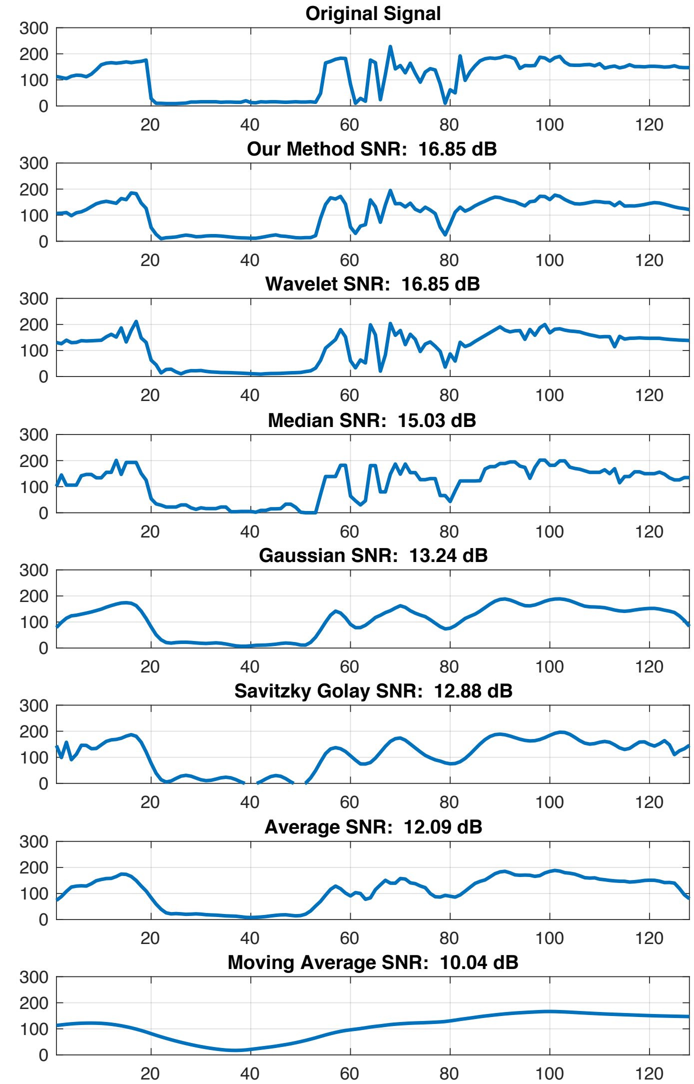


Fig. 10: Comparison of different smoothing techniques where y axis represents intensity value and x axis represents row index

V. Conclusion

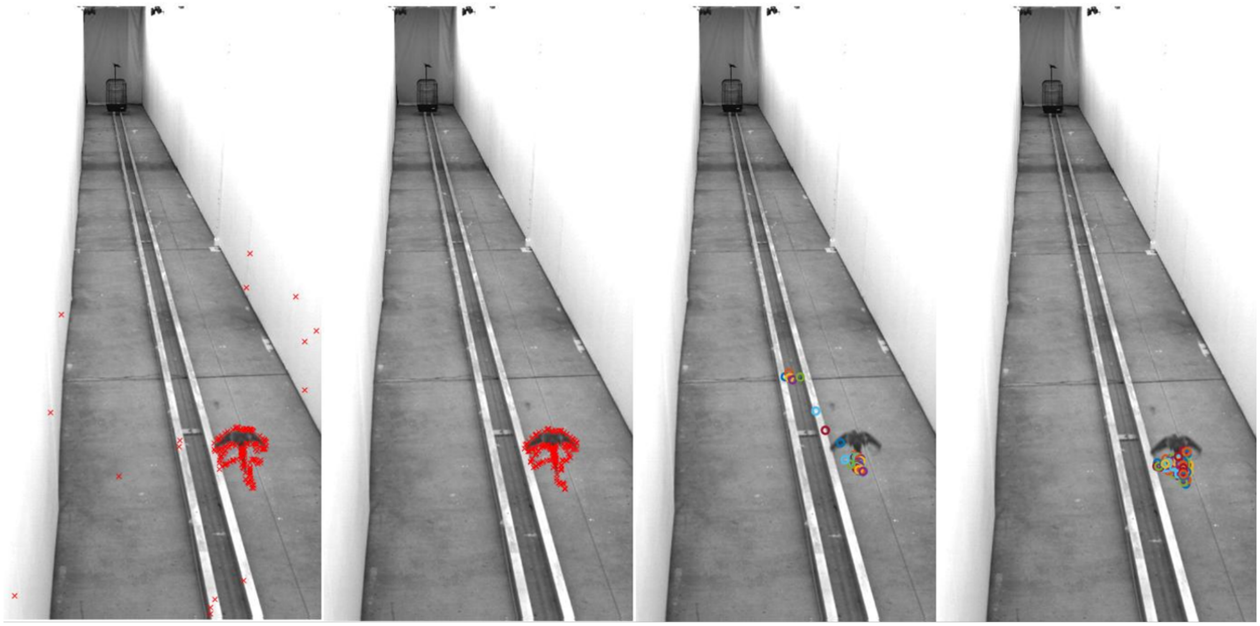
The image denoising technique described here is not limited to help with edge detection. It can be applied to a variety of problems in computer vision and medical image analysis. The main advantage of our denoising method is that it makes use of the high speed and the limited computational complexity of the WORM filters to reduce the noise. Most importantly the use of the dynamic mask overcomes the problem of blurring the edges.

Acknowledgment

This work was co-funded by the Australian Research Council and by Boeing Research & Technology Australia through a Linkage Project Grant (LP 130100483), and an ARC Distinguished Outstanding Researcher Award (DP140100914).

TABLE I: Detection accuracy with various types and densities of noise added to the images

Noise density (m/d)	0.1		0.2		0.3		0.4		0.5	
Noise type	SPN	GWN	SPN	GWN	SPN	GWN	SPN	GWN	SPN	GWN
Without Denoising	90%	92%	85%	86%	73%	69%	54%	48%	35%	32%
With Denoising	93%	94%	91%	89%	79%	75%	65%	62%	44%	37%



(a) (b) (c) (d)

Fig. 11: (a) Logic map for k=20 on raw frames 79 & 78 (b) Logic map for k=20 on denoised frames 79 & 78 (c) Centroids of different logic maps on raw frames 79 & 78 (d) Centroids of different logic maps on denoised frames 79 & 78

REFERENCES

[1] I. Schiffner and M. V. Srinivasan, "Direct Evidence for Vision-based Control of Flight Speed in Budgerigars," *Scientific Reports*, vol. 5, p. 10992, 2015.

[2] H. D. Vo, I. Schiffner, and M. V. Srinivasan, "Anticipatory manoeuvres in bird flight," *Sci Rep*, vol. 6, p. 27591, Jun 2016.

[3] M. V. Srinivasan, "Visual control of navigation in insects and its relevance for robotics," *Current Opinion in Neurobiology*, vol. 21, no. 4, pp. 535–543, 2011.

[4] H. Guo, J. E. Odegard, M. Lang, R. A. Gopinath, I. W. Selesnick, and C. S. Burrus, "Wavelet based speckle reduction with application to SAR based ATD/R," in *Proceedings - International Conference on Image Processing, ICIP*, vol. 1. IEEE Comput. Soc. Press, 1994, pp. 75–79.

[5] R. D. Nowak, "Wavelet-based Rician noise removal for magnetic resonance imaging," *IEEE Transactions on Image Processing*, vol. 8, no. 10, pp. 1408–1419, 1999.

[6] M. Lindenbaum, M. Fischer, and A. Bruckstein, "On Gabor's contribution to image enhancement," *Pattern Recognition*, vol. 27, no. 1, pp. 1–8, jan 1994.

[7] J. N. Lin, X. Nie, and R. Unbehauen, "Two-Dimensional LMS Adaptive Filter Incorporating a Local-Mean Estimator for Image Processing," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 40, no. 7, pp. 417–428, jul 1993.

[8] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filters under structural constraints," *1993 IEEE International Symposium on Circuits and Systems*, vol. 942, no. 2, pp. 942–945.

[9] K. E. Barner, "Rank Conditioned Rank Selection Filters for Signal Restoration," *IEEE Transactions on Image Processing*, vol. 3, no. 2, pp. 192–206, mar 1994.

[10] A. B. Hamza, P. L. Luque-Escamilla, J. Martínez-Aroza, and R. Román-Roldán, "Removing noise and preserving details with relaxed median filters," *Journal of Mathematical Imaging and Vision*, vol. 11, no. 2, pp. 161–177, 1999.

[11] L. A. Barford, R. S. Fazzio, and D. R. Smith, "An introduction to wavelets," *Hewlett-Packard Labs, Bristol, UK, Tech. Rep. HPL-92-124*, vol. 2, no. 2, pp. 1–29, 1992.

[12] "Wavelet Toolbox Reference." [Online]. Available: https://au.mathworks.com/products/wavelet.html

[13] D. Karmaker, I. Schiffner, M. Wilson, and V. Mandyam, "The bird gets caught by the WORM : tracking multiple deformable objects in noisy environments using Weight ORdered logic Maps," in *International Symposium on Visual Computing (ISVC)*. Springer-Verlag, 2018 in press.