inquiro
UNDERGRADUATE RESEARCH JOURNAL
of The University of Alabama at Birmingham

THE UNIVERSITY OF ALABAMA AT BIRMINGHAM
Knowledge that will change your world

An Adaptive Kernel-Growing Median Filter for High Noise Images

Jacob Laurel

Department of Electrical and Computer Engineering, University of Alabama at Birmingham, Birmingham, AL, USA

Electrical and Computer Engineering Departmental Honors Program, University of Alabama at Birmingham, AL, USA

ABSTRACT

A novel adaptive median filter is presented that can restore images corrupted by salt and pepper noise levels greater than 90%. The algorithm operates by adapting to the amount of available visual data in the image by iteratively increasing the size of the median kernel. The algorithm then detects the edges and reruns the adaptive median filtering process on just those edge pixels to improve edge consistency. Lastly, post-processing is done on the image using the Perona-Malik diffusion process for smoothing and an Unsharpen filter to improve contrast. The results of our algorithm show root-means-quare error improvement of the reconstruction compared to the state-of-the-art filter for image reconstruction.

KEYWORDS: adaptive-median filter, impulse noise, kernel, reconstruction

INTRODUCTION

Noise is one of the main problems faced in any kind of signal or image reconstruction process. One very common type of noise that arises in natural images is impulse noise, commonly referred to as “salt and pepper” noise[1]. Impulse noise manifests as image pixels (picture elements) that spike in color or intensity relative to the surrounding pixels; i.e. if part of an image is mostly green pixels, a random bright white pixel would be considered impulse noise.

Impulse noise at lower levels can be filtered out rather trivially by a simple median filter, but at ultra-high levels, i.e. greater than 80% of the image is affected by noise, the image reconstruction process is significantly harder. Furthermore, since color images consist of 3 channels (red, green and blue) the noise also exists in all 3 domains, and, as such, it must be filtered out on each domain, which adds to the complexity of the problem. When images are strongly corrupted, (i.e. a radio astronomical image that has been subjected to atmospheric noise), only a small portion of their pixels are truly correct, which means the amount of accurate data that can be used to reconstruct the corrupted noise pixels is sparse and thus not completely reliable. Various methods have been proposed to handle this problem, though median filters and interpolation methods have traditionally been some of the main techniques used due to their robust nature[2,3]. Linear filters make use of 2-D convolution in which n×n mask of pixels (called the kernel) is swept across the image and the central location in the kernel corresponds to the specific pixel that is being filtered. An illustration of this concept can be seen below in Figure 1.

[Figure 1: See "An adaptive kernel growing …" *PDF]*

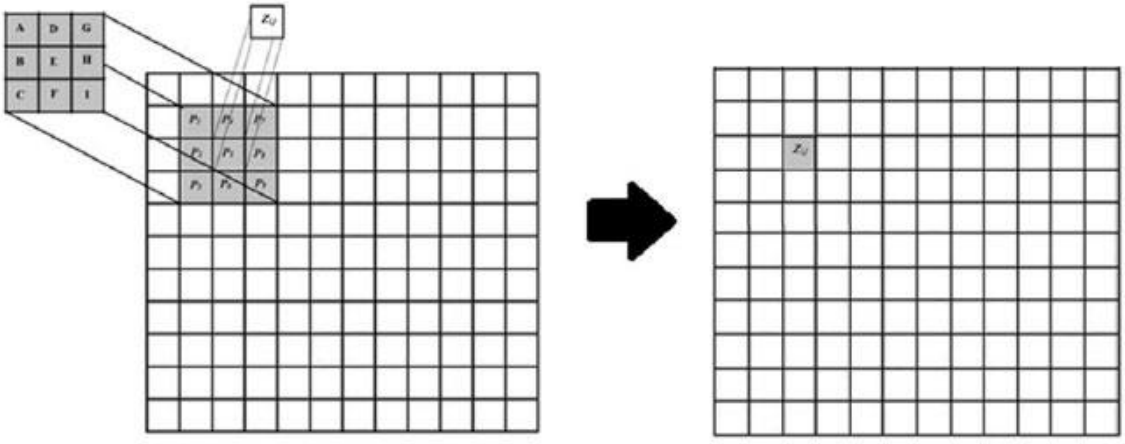

Figure 1 | A 3 × 3 filter with output pixel $Z_{i,j}$

Each pixel location where the kernel intersects the image is the product of the kernel value at that location with the corresponding actual image pixel value. This gives n×n different products, and the summation of all these products is used to replace the pixel in the image that corresponds with the central pixel in the kernel, thus performing the filtering on that pixel. This concept is mathematically shown in (1).

$$Z_{i,j} = A(P_1) + B(P_2) + C(P_3) + D(P_4) + E(P_5) + F(P_6) + G(P_7) + H(P_8) + I(P_9) \quad (1)$$

The median filter is slightly different, in that the central pixel is replaced with the median of the pixels in the kernel instead of the weighted sum. Recently novel additions to the median filter have been implemented that employ a variety of concepts, such as adaptiveness, fuzzy logic, or dynamic programming[4,5]. This paper will present the novel application of a new type of adaptive-median filter that has been shown to robustly reconstruct high-noise images to a very high level of accuracy. This paper will then compare the results of the novel adaptive-median filter to existing interpolation methods.

METHODS AND IMPLEMENTATION

Several images were obtained in order for the algorithm to be tested. Each image was corrupted to have impulse noise levels of 90%, 95% and 98% by randomly setting that percentage of the image's pixels to have either the minimum or maximum possible intensity value. Each image was then separately processed. First, noise pixels were located in the image by identifying pixels of minimum or maximum intensity, and then set to zero. The adaptive median filter ran specifically on these zero-valued pixels for each of the 3 pages of the image. The kernel size for the initial iteration of the adaptive median filter was set to 3 × 3. If a pixel was still found to have a value of zero after the first iteration, its position would be saved and then another

iteration of the median filter would be run, but this time the size of the kernel would be increased to 5 × 5. Likewise, this process of increasing the length and width of the kernel by 2 pixels was done until all of the zero-valued pixels were fully restored.

Upon achieving a reconstruction of the image, the object and feature edge pixel locations were detected using a standard deviation filter that was run on each page of the image. This filter calculates the standard deviation of all pixels in the kernel, and then replaces the central value of the kernel with that standard deviation value. This can be seen in (2), where x_i corresponds to each element in the filtering kernel and μ corresponds to the average of all the pixels' intensities. M corresponds to the number of elements in the kernel; so for a 3 × 3 kernel, M would be equal to 9.

$$Standard\ Deviation = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(x_i - \mu)^2} \quad (2)$$

This is useful since image regions that contain edges will have high local standard deviations. Upon detecting the edges by thresholding the image intensity, the width of these edges was then increased using a single iteration of binary morphological dilation followed by morphological closing. After performing these morphological operations the location of each edge pixel was saved. The same adaptivemedian filtering process previously mentioned was then run specifically on these edge pixels.

After fully reconstructing the edge pixels over the course of multiple iterations of the adaptive-median filter, the image was then processed to improve the appearance. This "post-processing" step first required the use of Perona-Malik anisotropic diffusion on the image. Perona-Malik anisotropic diffusion is the process of selectively blurring an image at different

regions, such as edges, based on a specified "conductance function" that has several parameters6. This anisotropic diffusion is based on the heat equation with a variable conductance function, denoted as *c(x,y,t)*, which can be seen below in (3).

$$\frac{\partial I}{\partial t} = c(x, y, t)\Delta I + \nabla c \cdot \nabla I \tag{3}$$

The conductance function chosen specifically for this algorithm can be seen below in (4).

$$c||\nabla I|| = e^{-\left(\frac{||\nabla I||}{K}\right)^2} \tag{4}$$

For the Perona-Malik diffusion, the conductance parameter *K* was set to 1000. Upon completion of the Perona-Malik diffusion process, the image contrast was then sharpened using MATLAB's built-in unsharpen filter in which the standard deviation of the Gaussian low-pass filter was by default set to 1, and the contrast threshold was set to 0.

Lastly, each fully processed and reconstructed image was compared to the original uncorrupted image to see if the algorithm could accurately recreate the images for each noise level. A function for root-mean-square error that relates how close the reconstructed and original images truly are was defined as follows, and was then calculated for the red, green and blue image channels.

$$Error_{RMS} = \sqrt{\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}(A_{i,j} - B_{i,j})^2} \tag{5}$$

$A_{i,j}$ and $B_{i,j}$ represent the pixels in the i^{th} row, j^{th} column of the original image and reconstructed image, respectively.

For comparison, a triangulation based interpolation image reconstruction method was then applied to the same images with the same noise levels, since current literature has made extensive use of this technique due to its perceived speed improvement[3]. This comparison was

done using MATLAB's built-in scattered interpolation function, and the root-mean-square error between the output reconstructed images and the originals was again calculated. The root-mean-square error of our adaptive-median filter algorithm was then compared to this error to see if our adaptive-median filter provided any advantage to the existing interpolation reconstruction methods.

MATERIALS

For the experiment, MATLAB R2015a in combination with Microsoft Excel were used for the computation. The MIT SUN database was used to obtain the test images[7]. All computations were carried out on a desktop computer with an AMD A10 quad core processor and 8.0 GB of RAM.

RESULTS

Images were chosen from the MIT SUN database for testing our algorithm[7]. For each test image and each noise level, the noisy image, the reconstructed image obtained from our adaptive-median filter, and the comparison reconstructed image obtained from standard interpolation can be seen in Figures 2 through 4. It is worth noting that for illustrative purposes four noises levels (30%, 90%, 95%, and 98%) are shown in Figures 2 through 4, but only the last three levels (90%, 95%, and 98%) were analyzed for results. Graphs displaying the RMS error, as a function of raw pixel intensity, for each image and noise level, that compare our novel adaptive-median filtering algorithm to the existing interpolation method are shown in Figures 5 through 7. A table showing the percentage of pixels restored after each iteration of the adaptive-median filtering algorithm, for each noise level and image, can be seen in Table 1.

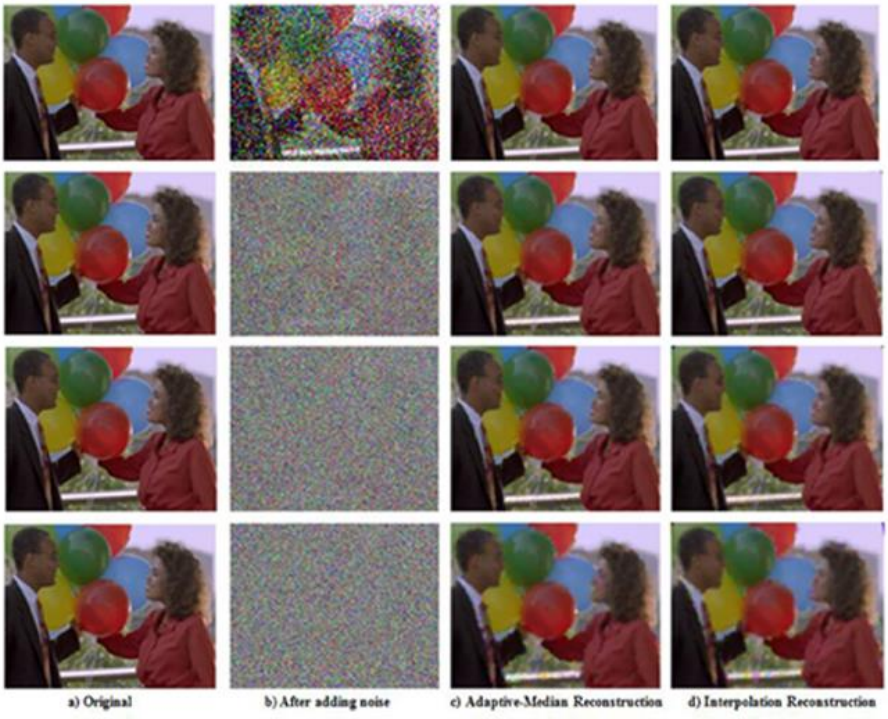

Figure 2 | The reconstructed "balloons" image comparison. *The noise levels in column b are 30%, 90%, 95%, and 98%, from top to bottom.*

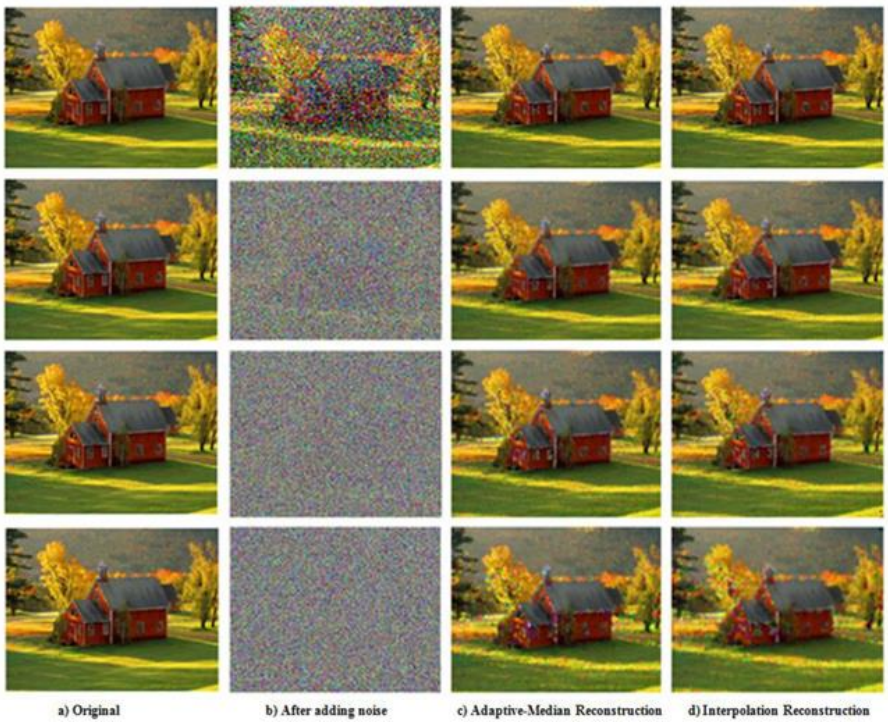

Figure 3 | The reconstructed "farm" image comparison. *The noise levels in column b are 30%, 90%, 95%, and 98%, from top to bottom.*

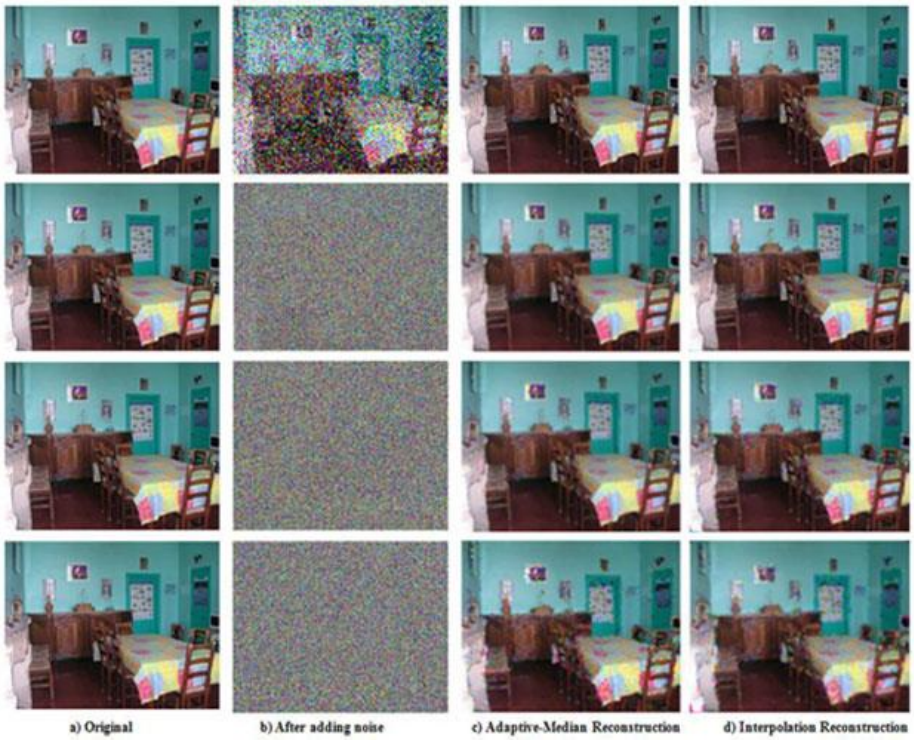

Figure 4 | The reconstructed "room" image comparison. *The noise levels in column b are 30%, 90%, 95%, and 98%, from top to bottom.*

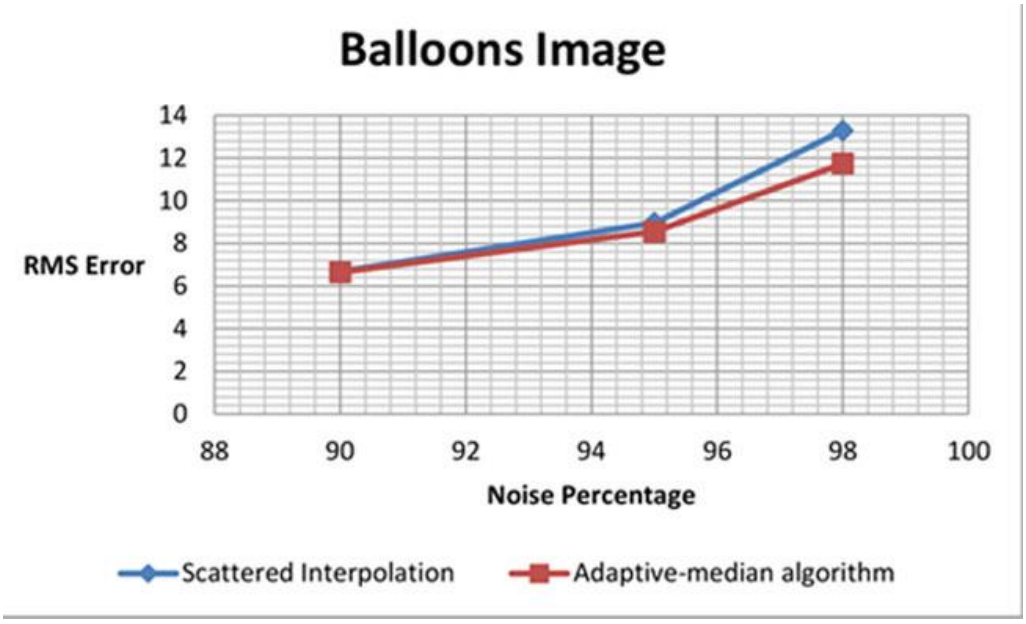

Figure 5 | Performance curve for the proposed algorithm and standard interpolation method for the balloons image.

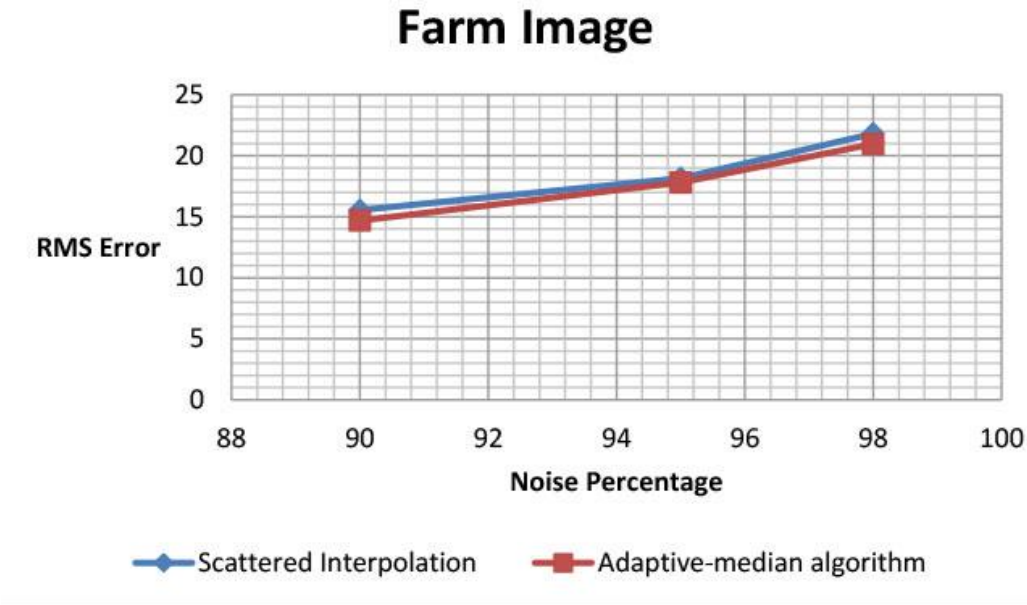

Figure 6 | Performance curve for the proposed algorithm and standard interpolation method for the farm image.

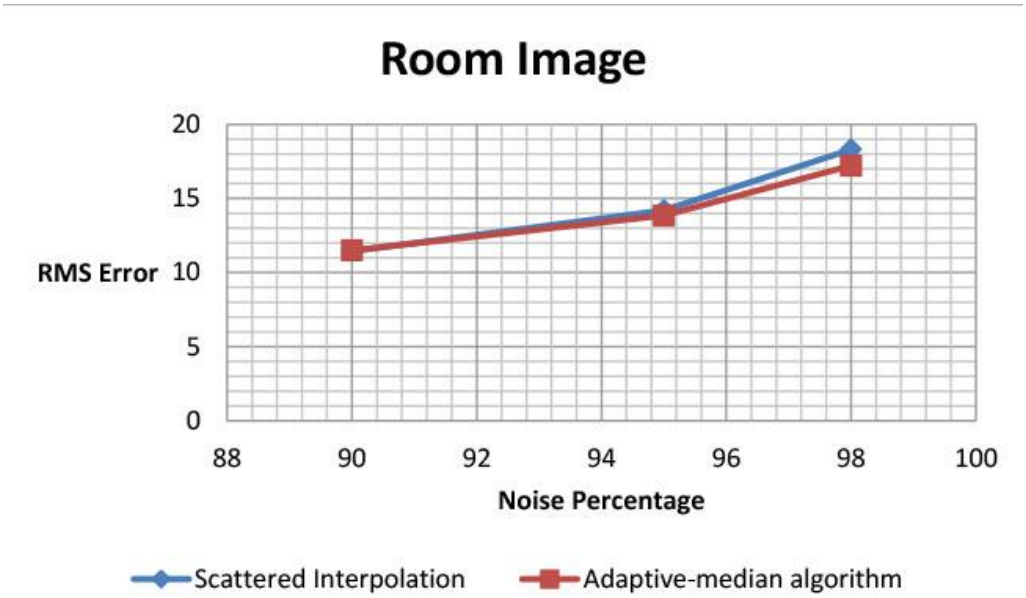

Figure 7 | Performance curve for the proposed algorithm and standard interpolation method for the room image.

Iteration	Balloons			Farm			Room		
	90% noise	95% noise	98% noise	90% noise	95% noise	98% noise	90% noise	95% noise	98% noise
0	9.965	4.974	1.991	9.724	4.8356	1.959	9.948	4.996	1.987
1	60.96	36.768	16.506	59.8252	35.8247	16.286	60.896	36.929	16.478
2	99.260	91.534	62.510	99.1118	90.5767	61.692	99.310	91.763	62.219
3	100	99.960	96.514	100	99.9476	96.150	100	99.976	96.314
4	100	100	100	100	100	100	100	100	100

Table 1 | Percentage of pixels restored for each iteration of the proposed algorithm for each image and noise level.

DISCUSSION AND FUTURE DIRECTIONS

Looking in Figures 2 through 4, it can be seen that for higher levels of noise, specifically the last two rows, our proposed algorithm provides an image that contains less visual artifacts when compared to the existing interpolation method. The interpolation method erroneously creates many false spots of color in the corners, as can be seen in Figure 2, column d. Our proposed algorithm also was better at maintaining contrast in the image, since the interpolation method blurred many of the key image details, such as the tablecloth in Figure 4, column d. It can also be seen from Figures 5 through 7 that our error curve (red) is slightly lower than the interpolation-based method's error curve (blue), thus showing that our algorithm successfully achieved lower RMS error. Our method provides a noticeable improvement on existing reconstruction methods for images, and computationally, ran just as fast as the existing methods; in some cases, ran noticeably faster, and thus has broad applications for this field. Future work will focus on trying to improve object edge textures, since these edges were the image regions that presented the most difficulties in accurate restoration.

REFERENCES

1. Zhengyang, G. & Le, Z. Improved adaptive median filter. *2014 Tenth International Conference on Computational Intelligence and Security (CIS)*, 44–46 (2014).
2. Li, Z., Zhang, W. & Lin, W. Adaptive median filter based on SNR estimation of single image. *2012 International Conference on Computer Science Service System (CSSS)* 246–249 (2012).
3. Kalyoncu, C., Toygar, O. & Demirel, H. Interpolation-based impulse noise removal. *IET Image Processing*, **7**(8), 777–785 (2013).
4. Xu, H. & Yue, X. An adaptive fuzzy switching filter for images corrupted by impulse

noise. *Sixth International Conference on Fuzzy Systems and Knowledge Discovery, 2009. FSKD '09* **3**, 383–387 (2009).

5. Yu, T.-H. & Mitra, S. K. Detail-preserving impulse noise removal of images using modified dynamic programming. *1990 International Conference on Acoustics, Speech, and Signal Processing, 1990. ICASSP-90* **4**, 1881–1884 (1990).
6. Perona, P. & Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(7), 629–639 (1990).

Xiao, J. *et al.* Sun database: Large-scale scene recognition from abbey to zoo. *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on* 3485–3492 (IEEE, 2010).