https://doi.org/10.2352/issn.2169-2629.2021.29.323

Long range diffusion with control of the directional differences

Ali Alsam, Hans Jakob Rivertz; Norwegian University of Science and Technology, Trondheim, Norway

Ali.Alsam@ntnu.no
Hans.J.Rivertz@ntnu.no

Abstract

A fast, spatially adaptive filter for smoothing colour images while preserving edges is proposed. To preserve the edges, we use a constraint that prohibits the increasing of the gradients in the process of diffusion. This constraint is shown to be very effective in preserving details and flexible in cases where more smoothing is desired. In addition, a filter of exponentially increasing diameter is used to allow averaging non-adjacent pixels, including those separated by strong edges.

Introduction

In imaging technology, noise is an integral part of the capturing devices and the physical fluctuations therein. On a more fundamental level, it is part of sampling a light signal that exhibits variations in the number of incoming photons. The physics of sensors and photons, as well as our ability to measure, means that some level of denoising is essential. Noise removal is important whether it is in medical imaging [1] where we are trying to make a diagnosis based on the recorded image or in digital photography [2] where the focus is on a pleasing visual result.

Mathematically, a noisy image is represented by a sum $y = x + n$, where y is the captured image, x is the desired noiseless image and n is noise [3]. Based on this formulation, we can think of noise reduction as a set of mathematical operations that aim to remove n from y to recover an estimate of the true image x. Indeed, what differentiates image denoising algorithms is the chosen set of mathematical operations to recover the image x and the arguments stated for their optimality.

When given a block of noisy pixels measurements, we might say that an optimal algorithm reduces the variance of the pixel values and proceeds to smooth the block by replacing those values with their average. A native averaging operation would indeed reduce the variance of the block. Thus, therefore, being the most optimal solution for the parts of the image that correspond to uniform regions in the original scene. Representing the pixel values with the average of their neighbours in a given block, image regions with edges and texture would be rendered unrecognisable. The larger the block, the less satisfying the result would be. In the previous decades, the focus of image denoising algorithms has been on reducing image variability while preserving edges and texture: an aim that has sprouted a plethora of different algorithms (add references). Today, there are many algorithms that preserve edges and other features while reducing noise, each with its advantages, disadvantages and computational complexity.

In this article, we combine two different approaches to image denoising. The first is the set of gradient domain algorithms that are mathematically similar to the landmark work of Perona, Malik [4]. Local gradients of the image are analysed and image denoising is cast as an iterative anisotropic diffusion process that prohibits smoothing the image data in the direction of strong gradients. In this family of algorithms, the permitted direction of smoothing is defined either by the strength of the individual gradients or an eigenvector analysis of the local tensor matrix. Unfortunately, these algorithms are unable to diffuse similar image values that are far from each other or separated by strong edges. The second set of algorithms searches the image for blocks that are statistically similar and averages their values. As an example of these algorithms, we consider non-local means and the work of [5]. Non-local means searches the image in a given sampling window for similar blocks that are then averaged to obtain the denoised image. Similarly, the algorithm presented in [5], stacks similar patches in a 3D-buffer and applies a filter in the wavelet domain. We note that although generally very effective in reducing noise, these algorithms will have problems with unique image features that do not repeat [3].

To elucidate our contribution, we consider the example of an image consisting of a black and white checker pattern with noise. In such an example, gradient domain denoising algorithms would diffuse the values inside each block separately. It would have been intuitive to average the pixels of the set of all the similar blocks to obtain two distinct values: one for white and another for black. On the other hand, non-local means would average the similar blocks but without analysing the gradients. To harness the power of the two approaches, we made use of the edge preserving constraint presented in [6, 7], where image diffusion was permitted in directions that do not lead to an increase in the local gradients. In their work,[6, 7], the authors considered diffusion from, and to, the centre pixel in a 3×3 block. Diffusion in any given direction was permitted when it didn't cause the gradients to increase in the other directions. In this paper, we keep this condition but we consider the diffusion from pixels outside the core 3×3 block. In the method section, we present a filter, of an exponentially increasing size where the values of selected pixels across the whole image can potentially be taken into consideration resulting in effective and fast denoising.

Method

Condition for edge conservation.

An RGB-colour image is represented as a function $\mathbf{I}$ defined on a rectangular grid in the plane with values (R, G, B) in $[0, 255]^3$. In all our calculations, the pixel values are mirrored along the boundaries of the image. Consider a source pixel $\mathbf{I}_s = \mathbf{I}(i_s, j_s)$, a target pixel $\mathbf{I}_t = \mathbf{I}(i_t, j_t)$ and the neighbour pixel values of $\mathbf{I}_t$, which we denote

$$\begin{array}{lll} \mathbf{I}_{n_3} = \mathbf{I}(i_t - 1, j_t - 1), & \mathbf{I}_{n_2} = \mathbf{I}(i_t, j_t - 1), & \mathbf{I}_{n_1} = \mathbf{I}(i_t + 1, j_t - 1), \\ \mathbf{I}_{n_4} = \mathbf{I}(i_t - 1, j_t), & & \mathbf{I}_{n_0} = \mathbf{I}(i_t + 1, j_t), \\ \mathbf{I}_{n_5} = \mathbf{I}(i_t - 1, j_t + 1), & \mathbf{I}_{n_6} = \mathbf{I}(i_t, j_t + 1), & \mathbf{I}_{n_7} = \mathbf{I}(i_t + 1, j_t + 1). \end{array} \tag{1}$$

Diffusion of $\mathbf{I}_s$ onto the target pixel $\mathbf{I}_t$ is controlled by watching the change of the differences caused by a test diffusion. A temporary target pixel value is computed $\mathbf{I}'_t = d\mathbf{I}_s + (1 - d)\mathbf{I}_t$ where $d = 0.3$. Diffusion from $\mathbf{I}_s$ to $\mathbf{I}_t$ is admissible if

$$\frac{\|\mathbf{I}_n - \mathbf{I}'_t\|^2 - \|\mathbf{I}_n - \mathbf{I}_t\|^2}{d} \leq \alpha \tag{2}$$

for all $n = n_0, n_1, \ldots, n_7$, where $0 \leq \alpha$. Condition (2) can be rewritten as

$$d\|\mathbf{P}_s\|^2 - 2\mathbf{P}_s \cdot \mathbf{P}_n \leq \alpha \tag{3}$$

where $\mathbf{P}_n = \mathbf{I}_n - \mathbf{I}_t$ and $\mathbf{P}_s = \mathbf{I}_s - \mathbf{I}_t$.

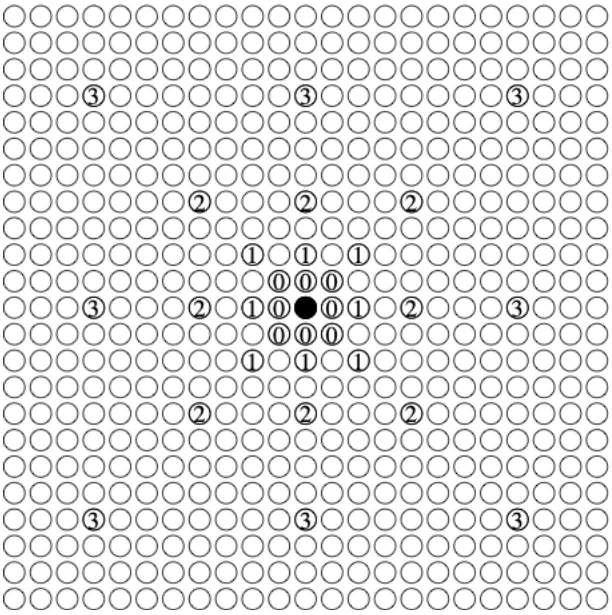


Figure 1: The grid shows the anatomy of the filter. The numbers indicate the non-zero elements of the generating kernels and the order in which they are applied.

Combining several source pixels

Given a set of source pixels $\mathbf{I}_s$, $s \in S_t$. Let $A_t \subset S_t$ be the set of indices s such that diffusion from $\mathbf{I}_s$ onto the target pixel $\mathbf{I}_t$ is admissible. The new target pixel value is obtained by:

$$\mathbf{I}_t + \sum_{s \in A_t} \gamma_s \mathbf{P}_s, \qquad (4)$$

where $\gamma_s > 0$ are appropriate diffusion factors with $\sum_{s \in S_t} \gamma_s < 1$. In [7], only the neighbouring pixels with distance one from the centre were used as source pixels. For each pixel $\mathbf{I}_t$, we take a set S_t of all pixels within some radius from $\mathbf{I}_t$. Any pixel from S_t can now be applied in the diffusion process, as long as its value does not breach the constraint outlined in equation (3). If the radius is one pixel, then the diffusion is identical to that described in [7]. Increasing the radius has obvious advantages as more pixels would be considered. Further, to allow closer pixels to influence the result of the diffusion more, we could use a two dimensional Gaussian distribution. Intuitively allowing diffusion over longer distances would reduce noise more effectively. Given the formulation of our constraint and considering all the pixels within a large radius is, however, computationally very expensive. We address this by using a varying range convolution process.

The varying range smoothing process

We define an iterative smoothing process. In each iteration we use eight source pixels for each target pixel t. The eight pixels are considered recursively. The source pixels in the first iteration are the neighbour pixels $S_{t,1} = \{n_0, n_1, \ldots, n_7\}$. In the subsequent iterations, the distances from the source pixels to the target pixel are doubled. Figure 1 shows the locations of the source pixels in the first four iterations. Here, the target pixel is marked as a black dot and the numbers, 0,1,2,3 correspond to the iteration number. For each iteration, we use the diffusion formula (4) with $\gamma_s = 1/8$ in the vertical and horizontal directions and $\gamma_s = 1/16$ in the diagonal direction.

Comparison with convolution methods

When there are no restrictions, that is $A_t = S_t$ in each iteration, the recursive process corresponds to a convolution filter with kernel: $G_0 = \frac{1}{16}\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix}$. The second step in an unrestricted process is equivalent to applying a convolution filter with kernel:

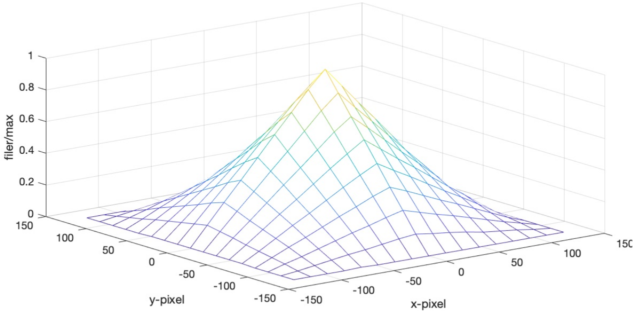


Figure 2: The effective filter after 7 iterations. The size of the filter is doubled for each iteration. Therefore, the filter is very efficient in smoothing large uniformly coloured areas.

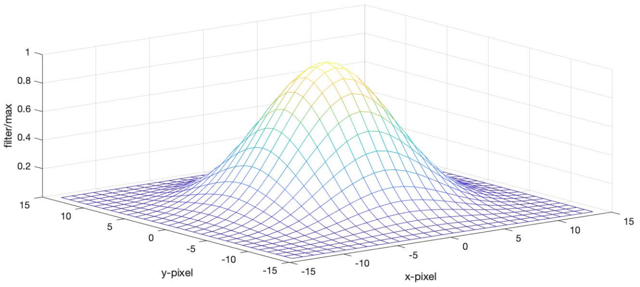


Figure 3: The effective filter after 3 iterations repeated twice. The kernel is $G^{*3} * G^{*3}$. Notice the Gaussian like shape of the kernel.

$$G_1 = \frac{1}{16}\begin{pmatrix} 1 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 4 & 0 & 2 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 1 \end{pmatrix}.$$

The third step of the process corresponds to using a kernel G_2 etc., where G_i is obtained from enlarging G_{i-1}. We enlarge G_{i-1} by adding a zero-row between each row and thereafter, adding a zero-column between each column, which corresponds to an exponentially growing kernel. Here, we underline that each pixel value is considered individually, i.e. we don't convolve the image with the filter.

Two unrestricted iterations give effectively a filter with kernel:

$$G_1 * G_0 = \frac{1}{256}\begin{pmatrix} 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 4 & 8 & 12 & 16 & 12 & 8 & 4 \\ 3 & 6 & 9 & 12 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 6 & 4 & 2 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \end{pmatrix}.$$

Without restrictions, n iterations are equal to the single convolution filter $G^{*n} = G_{n-1} * \cdots * G_0$. The coefficients of G^{*n} are:

$$G^{*n}_{i,j} = \begin{cases} \frac{(2^n - |i|)(2^n - |j|)}{16^n}, & \text{when} |i|, |j| < 2^n \\ 0 & \text{otherwise.} \end{cases}$$

Figure 2 shows the normalised graph of G^{*7}. The form of the filter looks like a pyramid. Repeating the unrestricted process twice is equivalent to a filter that is very similar to a Gaussian. Without any restrictions on the diffusion, these filters will blur the image rapidly.

Complexity

A direct convolution with G^{*n} requires $16 \cdot 4^n - 8 \cdot 2^n + 1$ calculations for each pixel. Alternatively, one could build another filter of approximately the same size by repeating a 3×3 kernel 2^n times. That would take approximately $8 \times 2^{n+1}$ calculations per pixel. The latter would however be Gaussian and not directly comparable. Our method uses $8n$ calculations for each pixel.

Weaker conditions

Removal of salt and pepper noise fails if two neighbouring pixels are affected. This is because condition (2) fails in the direction between the two pixels. We add two weaker conditions.

Strong condition Diffusion $\mathbf{I}_8 \rightarrow \mathbf{I}'_8$ along the direction i is blocked iff $s\|\mathbf{P}_i^d\|^2 - 2\mathbf{P}_i^d \cdot \mathbf{P}_r^1 > \alpha$ for at least one $r = 0, 1, \ldots, 7$.

Medium condition Diffusion $\mathbf{I}_8 \rightarrow \mathbf{I}'_8$ along the direction i is blocked iff $s\|\mathbf{P}_i^d\|^2 - 2\mathbf{P}_i^d \cdot \mathbf{P}_r^1 > \alpha$ for at least two $r = 0, 1, \ldots, 7$.

Weak condition Diffusion $\mathbf{I}_8 \rightarrow \mathbf{I}'_8$ along the direction i is blocked iff $s\|\mathbf{P}_i^d\|^2 - 2\mathbf{P}_i^d \cdot \mathbf{P}_r^1 > \alpha$ for at least three $r = 0, 1, \ldots, 7$.

Results

Comparison with N.L.M.D.

In this section, we compare our method with the non-local means denoising (NLMD) [8, 9]. To obtain the results of denoising, using non-local means, the parrot image and the Lena image depicted in figures 4a and 6a respectively, were uploaded to the NLMD-demo website [10]. Furthermore, we use the demo algorithm available on the site to add noise with standard deviation $\sigma = 5$ and $\sigma = 10$ to the parrot image. Here we note that in contrast to the NLMD-method, our approach does require knowledge of the noise level. The images used were all 8-bits in the range 0-255. The results for the parrot image are shown in figure 4, where we note that for α values 50 and greater, the results obtained by the proposed algorithm are comparable to those achieved by non-local means.

When more noise is added to the image, we find that the performance of non-local means is slightly favourable. Additionally, a large α value is required for the proposed algorithm to render a comparable result. A numerical comparison between the different settings, and methods, is provided in table 1, where we find that non-local means is slightly better in terms of the peak signal to noise metric.

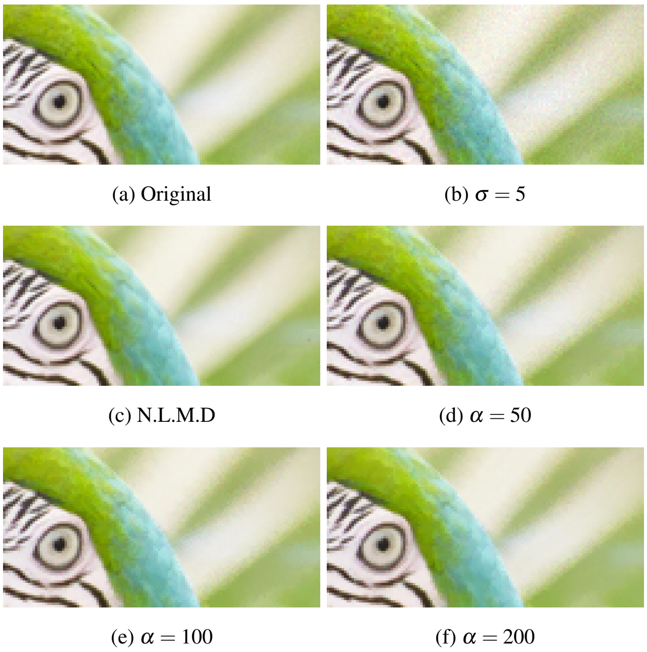
To evaluate the algorithm in the presence of excessive noise we add Gaussian white noise to the Lena image with $\sigma = 30$. Again, we compare our results with those obtained from non-local means. In this case, both algorithms perform well in removing most of the noise but non-local means is better in some image regions, which is reflected in the peak signal to noise metric tabulated in table 2. Here we note that non-local means samples a larger window than that used in our algorithm and it is designed to remove white noise.

Removal of salt & pepper noise

Having tested the performance of the method and its effectiveness at removing white noise, we experimented with a different type of noise, namely salt and pepper. Moreover, we explored the differences between the strong, medium and weak conditions defined in the weaker conditions subsection. Figure 7, depicts the result of the method applied on a female portrait from the Kodak database with salt & pepper noise added. The processing results are shown in Figure 7b, Figure 7c and Figure 7d. From Figure 7b we notice that all isolated colour dots are removed. In addition, Figure 7c clearly shows that all isolated colour dot pairs are removed. In all the resultant images, the original details are very well conserved.

Varying the diffusion factor

It is well understood that a high diffusion factor can lead to unstable results and, in the case of image smoothing, false edges. It is thus recommended to keep the diffusion factor low. To explore the effect of the level of diffusion on real images, we use the diffusion factor pair $(1/20, 1/40)$ instead of $(1/8, 1/16)$, which were used in

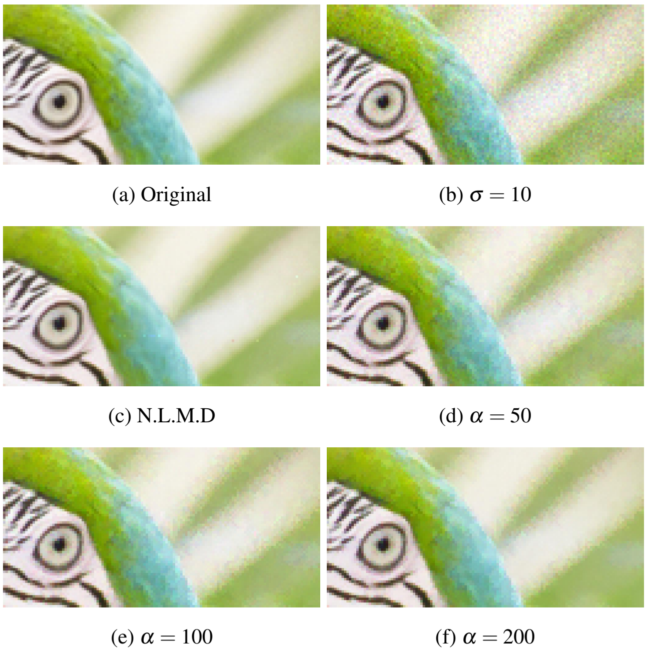


(a) Original (b) $\sigma = 5$

(c) N.L.M.D (d) $\alpha = 50$

(e) $\alpha = 100$ (f) $\alpha = 200$

Figure 4: The images show a segment of a larger parrot image. In image (b), noise with $\sigma = 5$ was added to the original. Image (c) shows the result obtained by using the NLMD-method. Images (d-f) show the results of our method with $\alpha = 50$, $\alpha = 100$ and $\alpha = 200$ respectively. Here we use the strong edge preservation condition.

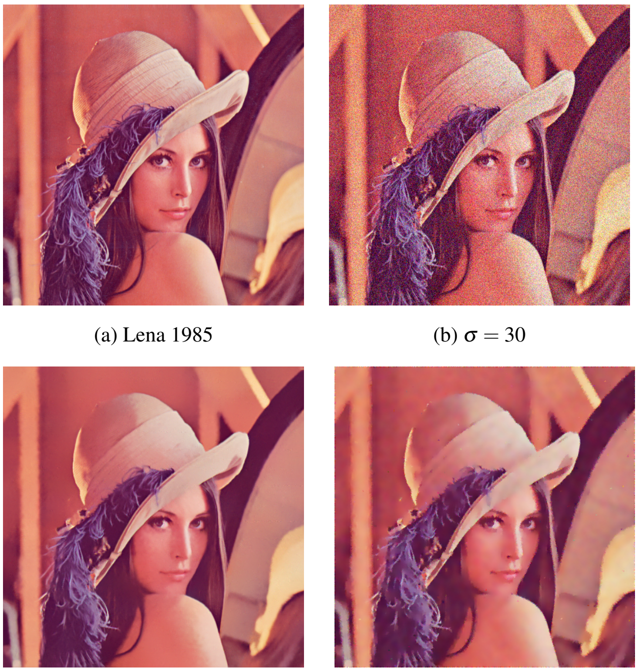


(a) Original (b) $\sigma = 10$

(c) N.L.M.D (d) $\alpha = 50$

(e) $\alpha = 100$ (f) $\alpha = 200$

Figure 5: The images show a section of the larger parrot image. In image (b), noise with $\sigma = 10$ is added. Image (c) shows the result obtained by the NLMD-method. Images (d-f) show the results of our method with $\alpha = 50$, $\alpha = 100$ and $\alpha = 200$ respectively.



(a) Lena 1985

(b) $\sigma = 30$

(c) NLMD

(d) Our method

Figure 6: Noise with $\delta = 30$ was added to the original. Denoising with NLMD, image c removes most of the visible noise. In our method, we use $\alpha = 800$. Also, we use the kernel G^{*3} twice with the strong condition. The diffusion factors are $1/8$ and $1/16$.

the previous experiments. We ran three experiments with $n = 7$ and $\alpha = 100$. To compensate for the reduced diffusion we ran the algorithm twice. The results of the experiments are shown in Figure 8. A closer inspection of the of the images shows effective noise reduction. In order to display the difference between the original and the algorithm's output, we calculate the absolute difference between the gradients of resultant image and the original image $\mathbf{I}$.

$$D(x,y) = \|\nabla \mathbf{I}(x,y) - \nabla \tilde{\mathbf{I}}(x,y)\|_1.$$

	$\sigma = 5$	$\sigma = 10$
Noisy	34.25	28.28
NLMD	40.18	37.06
$\alpha = 25$	34.37	32.89
$\alpha = 50$	36.59	33.67
$\alpha = 100$	37.66	34.20
$\alpha = 200$	37.74	33.36

Table 1: The PSNR between the original Parrot image and the processed images.

Noisy	18.85
NLMD	30.17
Our	28.41

Table 2: The PSNR between the original Lena image and the processed images.

Conclusion

We have presented an effective edge-preserving smoothing algorithm. By using an exponentially increasing size filter, the algorithm allows diffusion between non-connected image regions without blurring edges. To preserve edges, we use a constraint that prohibits the norm of the gradient from increasing during smoothing. Further, we show that this constraint can be loosened to allow more diffusion when needed. Our tests show that the algorithm is comparable to the effective non-local means denoising. In contrast to our algorithm, non-local means requires knowledge of the present noise statistics and works best for white noise. With an automatic adjustment of the parameters based on information about the noise in the picture, our method could be enhanced. We especially believe that the computational complexity of the constraint can be substantially reduced.

Salt & Pepper	25.37
Weak	37.22
Medium	36.16
Strong	34.53

Table 3: The PSNR between the original Woman image and the processed images.

Acknowledgement

Thanks to Solveig Rivertz for proofreading this paper.

References

[1] Diwakar, M., Kumar, M.: A review on ct image noise and its denoising. Biomedical Signal Processing and Control **42** (2018) 73–88

[2] Bertalmío, M.: Denoising of photographic images and video: fundamentals, open challenges and new trends. Springer Publishing Company, Berlin (2018)

[3] Fan, L., Zhang, F., Fan, H., Zhang, C.: Brief review of image denoising techniques. Visual Computing for Industry, Biomedicine, and Art **2** (2019) 7

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

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

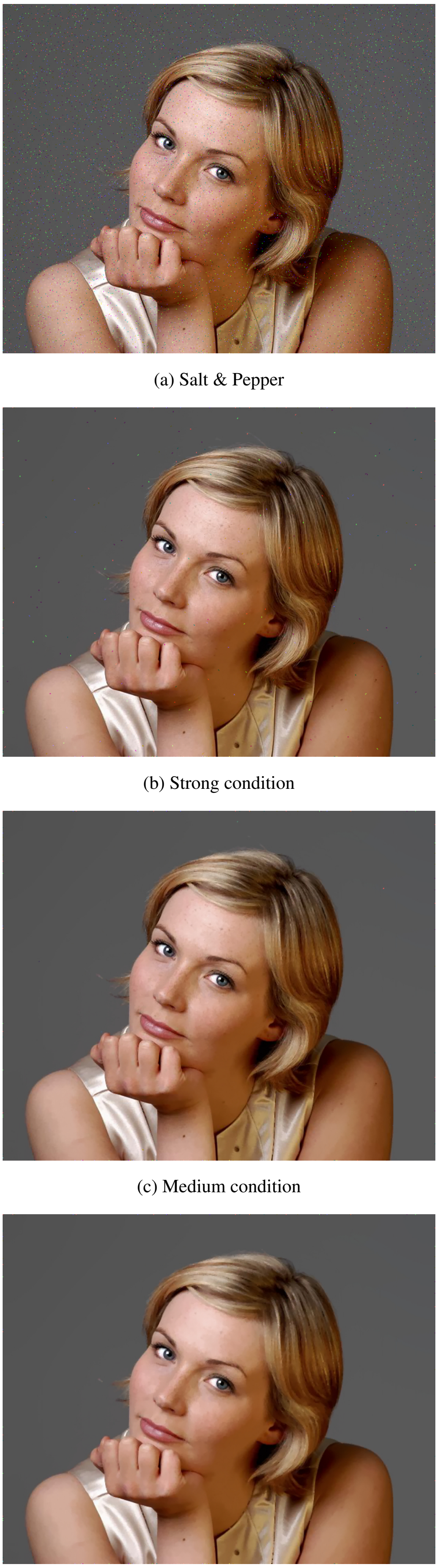
[6] Alsam, A., Rivertz, H.J.: Fast edge preserving smoothing algorithm. In: Proceedings of the 13th IASTED International Conference on Signal and Image Processing, ACTA Press (2011) 8–12

[7] Alsam, A., Rivertz, H.J.: Color edge preserving smoothing. In Bebis, G., et al., eds.: Advances in Visual Computing, Berlin, Heidelberg, Springer Berlin Heidelberg (2013) 79–86

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

[9] Buades, A., Coll, B., Morel, J.M.: Non-Local Means Denoising. IPOL Journal **1** (2011) 208–212 `https://doi.org/10.5201/ipol.2011.bcm_nlm`.

[10] Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. `https://ipolcore.ipol.im/demo/clientApp/demo.html?id=55` (2011) Accessed: 2021 June 7th.



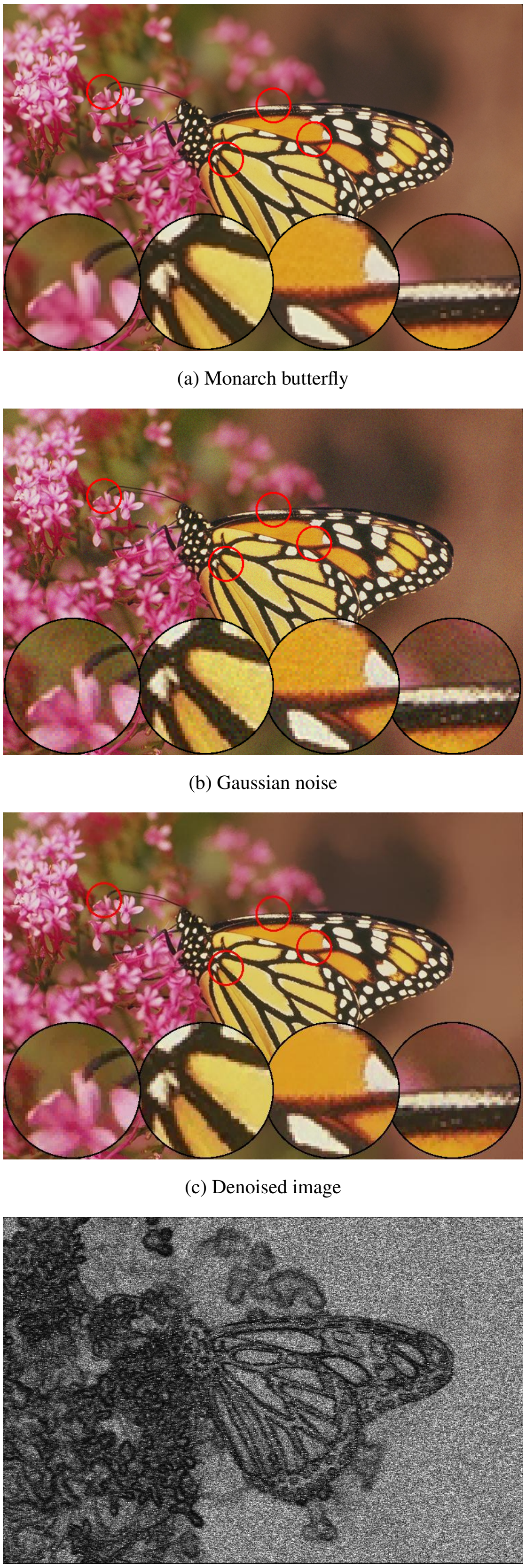
(a) Salt & Pepper

(b) Strong condition

(c) Medium condition

(d) Weak condition

Figure 7: Salt and pepper noise is added in each colour layer in image (a). We have used $\alpha = 25$. Diffusion to the centre pixel is blocked by using (b) the strong condition, (c) the medium condition and (d) the weak condition.



(a) Monarch butterfly

(b) Gaussian noise

(c) Denoised image

(d) Gradient difference

Figure 8: The original image (a) has Gaussian noise with mean 0 and variance 0.001 (b). We applied our method with a diffusion factor pair $(1/20, 1/40)$, $\alpha = 100$, and $n = 7$. We ran our diffusion process twice(c). Image (d) shows the absolute difference in the gradients between the image (b) and (c). Here we use the strong edge preservation condition.