https://doi.org/10.2352/CIC.2022.30.1.25

Three dimensional surface preserving smoothing

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

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

Abstract

We propose a new anisotropic diffusion process for removing noise from MRI images without distorting the edges. The method is based on a simple principle: any diffusion that increases a gradient at neighbouring pixels should be prohibited. From this principle, we deduce an inequality that allows diffusion along the edges but not across them. We introduce promising results using synthetic data with various types of noise as well as real MRI scans.

Introduction

The arrival of three-dimensional medical and industrial imaging, presented imaging researchers with the challenge of extending algorithms that are used in enhancing grey scale and colour images to the higher dimensional spaces, such as magnetic resonance imaging (MRI) and 3D scanning. One of the most important algorithms that we wish to extend is an efficient method to remove digital noise.

In the image processing literature, there are many noise models. There is Gaussian noise, electronic noise, white noise, Brownian noise, salt and pepper noise, periodic noise, quantisation noise, speckle noise, and structured noise. Noise is simply unavoidable, which is why all digital cameras are equipped with noise removal algorithms as part of the image production pipeline. Given the many types of noise and the number of digital images captured on a daily basis, it is to no surprise that the literature and commercial interest in the topic is vast.

Theoretically, a real image is defined as the sum of a noise free matrix and another that is pure noise. Given this definition the best noise removal method is one that retrieves the ideal image and separates it from the noise data. This ideal task is, however, difficult due to the similarity between noise and other high frequencies elements such as edges. Thus, the goal of noise removal can be reformulated to include the removal of noise while preserving texture, edge details and maintaining the integrity of region boundaries.

What is required is to remove high frequencies associated with noise while preserving those pertaining to edges and texture. To achieve this goal we need to recognise edges and smooth the data along but not across them. This task is achieved in landmark noise removal algorithms, including anisotropic diffusion which uses the principles of heat diffusion in mediums with different diffusion constants to develop an algorithm which starts by calculating the gradients in four directions: north, south, east and west. Finally, the image data is diffused in the direction that has the least resistance.

Other important algorithms include total variation [5, 8, 9], where the noise removal is cast as a minimisation problem of the integral of the gradients subject to the resultant that the noise free image is close to the original and that the noise is within a given standard divination. On the other hand, in bilateral filtering [4, 7, 10], unique filters are designed for each image pixel where both the spatial location of the pixels and the differences are taken into consideration.

Previously, the authors in [1–3], proposed an anisotropic diffusion method that is based on the constraint that image gradients should decrease in smoothing and any increase in gradients is due to edges. The main idea of this algorithm is that diffusion is permitted in directions that do not cause an increase in the derivative of the image, e.g. smoothing in the north direction is allowed only if it results in decreasing differences in the south, east and west directions.

In this paper, we apply the idea from [1–3] to synthetic three dimensional data with different types of added noise. We consider a given pixel and its twenty-six neighbours. We then examine the change in the derivatives when the pixel is averaged with a given neighbouring value. If the change is within an acceptable noise level, we permit diffusion. Otherwise diffusion is prohibited.

Our results show that the method removes noise if we choose the parameter that controls the diffusion carefully. This parameter was determined theoretically. Experiments show that the theoretic value of lambda was correct, by comparing two cases:

- First we used a parameter that was tuned to remove noise in regions with approximately uniform data values. The experiments did well in removing the noise in these regions, but failed in regions with sufficiently large gradients.
- Secondly we used a parameter that was made to remove noise in regions with relatively large gradients.

As expected from the theory, we saw that the noise in the high gradient areas was reduced in the second experiment and not in the first experiment. Our data was synthetic with added Gaussian and speckle noise.

The diffusion algorithm

The proposed algorithm is iterative where a data set is filtered in time and space. Without any constraints on the diffusion, the process will give a blurred dataset and all the intensity values will converge to their average. In earlier methods, there are many constraints that have the common aim of reducing the diffusion in directions of high gradients. Hence the idea is to preserve high gradients.

Our idea is the opposite. When a sharp difference is removed, some neighbouring gradients will increase and it is this increase which we prohibit. We consider the directional differences in 26 directions and how these are changed by diffusion in

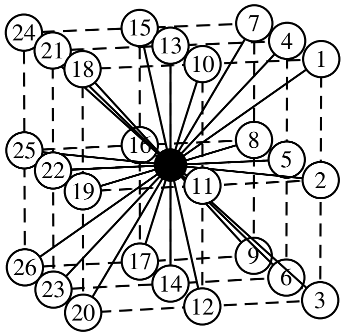


Figure 1. The 26 closest neighbours and the directions of diffusion and derivative.

each of the directions. Let us consider a pixel location and diffusion from its 26 neighbours as shown in Figure 1. We denote the 26 directions by the numbers 1 to 26. The directional differences are $d_i = I_i - I_0$, where $i = 1, 2, \ldots, 26$ and I_0 is the centre pixel. We will consider the 26 diffusion from I_j, where $j = 1, 2, \ldots 26$ to I_0. These are test diffusions and we consider them separately. Diffusion along direction $j = 1, 2, \ldots 26$ gives the new value of the centre pixel $I_0' = sI_j + (1-s)I_0$, where $s \in (0,1)$ is the test diffusion factor. The 26 directional differences have changed to $d_i' = I_i - I_0' = d_i - sd_j$.

Condition. *We will not allow a diffusion in direction* $j = 1, 2, \ldots, 26$ *if there is one direction i with*

$$\frac{{d_i'}^2 - {d_i}^2}{s} > \lambda, \tag{1}$$

where $\lambda > 0$ *is a small number.*

We replace d_i' with $d_i - sd_j$ in the inequality and rearrange (1) to the equivalent form $s{d_j}^2 - \lambda > 2d_i d_j$ The condition for allowing a diffusion in direction j is therefore that $s{d_j}^2 - \lambda \leq \min_i\{2d_i d_j\}$. If $d_j = 0$, the condition is satisfied since λ is positive. The two other possibilities $d_j < 0$ and $d_j > 0$ split the condition into two parts.

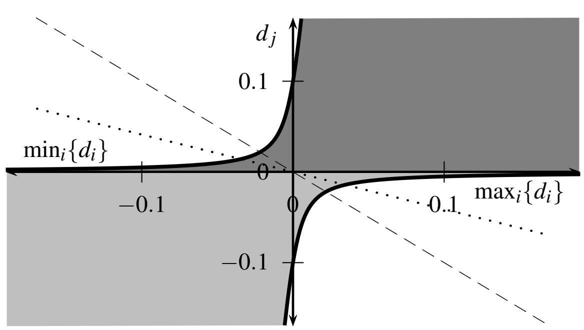


Figure 2. If $d_j > 0$, diffusion is allowed in the direction j, if the point $(d_j, \min_i d_i)$ lies in the dark grey area. If $d_j < 0$, then $(d_j, \max_i d_i)$ must be found in the light grey area. The parameters are $s = 0.1$ and $\lambda = 10^{-3}$.

Theorem 1. *The condition above is equivalent to the following: diffusion from* I_j *to* I_0 *is not allowed if*

$$sd_j - \lambda/d_j \leq 2\min_i\{d_i\} \text{ for } d_j > 0 \tag{2}$$

$$sd_j - \lambda/d_j \geq 2\max_i\{d_i\} \text{ for } d_j < 0. \tag{3}$$

Corollary. *The condition for diffusion is equivalent to allowing diffusion in the j-direction if and only if* $\max_i\{d_i\}d_j/s - \sqrt{\lambda/s + (\max_i\{d_i\})^2/s^2} \leq d_j \leq \min_i\{d_i\}d_j/s + \sqrt{\lambda/s + (\min_i\{d_i\})^2/s^2}$

Proof. We solve the inequalities (2) and (3): $d_j \leq \min_i\{d_i\}d_j/s + \sqrt{\lambda/s + (\min_i\{d_i\})^2/s^2}$ $d_j \leq \max_i\{d_i\}d_j/s - \sqrt{\lambda/s + (\max_i\{d_i\})^2/s^2}$, □

The light grey area in figure 2 shows the condition on d_j for allowing diffusion in the direction j when $d_j < 0$. The dark grey area in figure 2 shows the condition on d_j for allowing diffusion in the direction j when $d_j > 0$.

Theorem 2. *If* $\min_i d_i = -\sqrt{\lambda/(2+s)}$ *and* $\max_i d_i = \sqrt{\lambda/(2+s)}$, *then the condition in* (1) *is satisfied for all directions and we have ordinary diffusion.*

Proof. The dashed line in Figure 2 is the curve given by the equation $d_j = -\min_i d_i$ and equivalently the curve $d_j = -\max_i d_i$. These intersection points have values $d_j = \pm\sqrt{\frac{\lambda}{2+s}}$. □

The theorem guarantees that we have diffusion in all directions when $\min_i d_i$ and $\max_i d_i$ are the abscissa of the intersection between the curves and the dashed line in Figure (2). Theorem 2 can be used to give a value of λ so that we get an effective smoothing in areas where the original had a zero gradient.

In practice, there will be noise in areas with a gradient and we would like to remove that as well. In other words, we wish to smooth along a direction where $|d_j| < \sigma$ when $\max_i d_i < M$ and $\min_i d_i > -M$. That is, we need to find a value for λ so that the points $\pm(-M, \sigma)$ are on the curves in Figure 2. Namely,

$$s\sigma - \lambda/\sigma = -2M.$$

We solve this for λ.

Theorem 3. *When* $\lambda = s\sigma^2 + 2\sigma M$, *then the method will smooth in directions* j *where* $|d_j| < \sigma$ *in points with* $\max_i d_i < M$ *and* $\min_i d_i > -M$.

Experiments

Correct value of λ

We ran the algorithm on test data with three types of added noise.

- Gaussian noise with standard deviation 0.05 was added to the synthetic data. The result for the Gaussian noise experiment is shown in Figure 4.
- Speckle noise with variance 0.05^2 was added to the synthetic data. The result for the Speckle noise experiment is shown in Figure 5.
- We added salt and pepper noise with volume density 1/1000. The result for the Speckle noise experiment is shown in Figure 6.

With standard deviation $\sigma = 0.05$, we calculate λ by using theorem 2. We use the value $\lambda = (2+s)\sigma^2 = 0.0055$. This value of λ will give diffusion in all directions in areas where the original data has uniform shading. For maximum gradient $M = 0.20$ and $\sigma = 0.05$, Theorem 3 gives the ideal value $\lambda = +s\sigma^2 + 2M\sigma \approx 0.02$ for the threshold λ.

Running the experiment on the synthetic data.

The experiment was run with the values $s = 0.1$, $\lambda \in \{0.02, 0.0055\}$ with 5, 10 and 20 iterations for the data with Gaussian and speckle noise. We used $\lambda = 0.0055$ and $s = 0.1$ with 20 iterations for the data with added salt and pepper noise.

Experiment on MRI.

We ran our algorithm on an MRI data-set for a macaque monkey head, (Figure 6(a)). The data was produced by De Castro et al. [6]. We ran the algorithm with $\lambda = 10^{-4}$ and $\lambda = 10^{-5}$. Figure 6 show the result after 10, 25,

Results

Salt&Pepper

The experiments using salt and pepper noise show that isolated dots are removed easily as shown in Figure 5. However, connected dots are not removed at all. The explanation for this is simple: for isolated white points the pairs $(d_j, \min_i d_i)$ are in the third quadrant in Figure 2, for connected white dots, $(d_j, \min_i d_i) = (d_j, 0)$ lies at the ordinate axis in Figure 2.

Removal of Gaussian and Speckle noise

Figure 3 shows that the algorithm removed the Gaussian and Speckle Noise that we added to the synthetic data. The algorithm ran with 5, 10 and 20 iterations. There is a slight difference between the results for $\lambda = 0.02$ and $\lambda = 0.0055$. The images (c) to (h) in Figure 3 and 4 show that the noise is removed effectively in areas with no gradient for both values of λ. In regions with a gradient in the original, the images (f) to (h) shows a slightly better result for $\lambda = 0.02$ compared to $\lambda = 0.0055$. This is consistent with Theorem 2 and Theorem 3.

The Macaque Monkey MRI test.

Figure 6 shows that our algorithm removes noise without removing details and edges.

Conclusion

The method is effective in reducing the added Gaussian and Speckle noise from the data. The algorithm removes both Pepper and salt grains in the picture effectively as long as the grains are isolated. Two neighbouring salt grains are preserved by the method. The method interprets two neighbouring grains as a short edge.

Although the data in this paper are monochromatic MRI scans, the method will also work on for multichannel 3D data, such as movies where the third dimension is time. Our method can also be used in smoothing multi spectral images by viewing these as three dimensional data.

Synthetic test data

A function $I(x,y,z)$ was created to replace real MRI data. The synthetic data offer control over the performance of the algorithm. The data was produced by the following function:

$$f(x,y,z) = \cos^4 2\sqrt{r + \left(\sin\frac{x}{4} + \frac{3}{2}\sin\frac{y}{4} + 3\sin\frac{z}{4}\right)}$$

This function makes non-uniform shells with gradient surfaces. This picture is modified by successively inverting the image in three random chosen affine half spaces given by: $\mathbf{n}_i \cdot (\mathbf{x} - \mathbf{p}_i) > 0$, where

$$\begin{aligned} \mathbf{n}_1 &= (-0.567140, -0.100444, -0.236113), \\ \mathbf{n}_2 &= (-0.615264, 0.992138, -0.732168), \text{ and} \\ \mathbf{n}_3 &= (-0.547542, 0.074655, 0.638890) \end{aligned}$$

are normal vectors of the respective affine planes and

$$\begin{aligned} \mathbf{p}_1 &= (74, 65, 54), \\ \mathbf{p}_2 &= (245, 205, 108), \text{ and} \\ \mathbf{p}_3 &= (174, 180, 77) \end{aligned}$$

are points in the respective affine planes.

We let $x, y \in \{1, 2, \ldots, 256\}$ and $z \in \{1, 2, \ldots, 128\}$. The number r is the distance from (x,y,z) to the centre $(128, 128, 64)$ of the image.

In the experiment we added different types of noise to the synthetic data and ran the algorithm on the data. The data produced was smoothed with a filter with a Gaussian smoothing kernel with standard deviation $\sigma = 0.75$ to avoid anti aliasing.

References

[1] Ali Alsam and Hans Jakob Rivertz. Fast edge preserving smoothing algorithm. In *Proceedings of the 13th IASTED International Conference on Signal and Image Processing*, pages 8–12. ACTA Press, 2011.

[2] Ali Alsam and Hans Jakob Rivertz. Color edge preserving smoothing. In *Advances in Visual Computing, 9th International Symposium, ISVC 2013*, pages 79–86. Springer-Verlag Berlin Heidelberg, 2013.

[3] Ali Alsam and Hans Jakob Rivertz. Improved color edge preserving smoothing. In *Norsk Informatikkonferanse*, 2019.

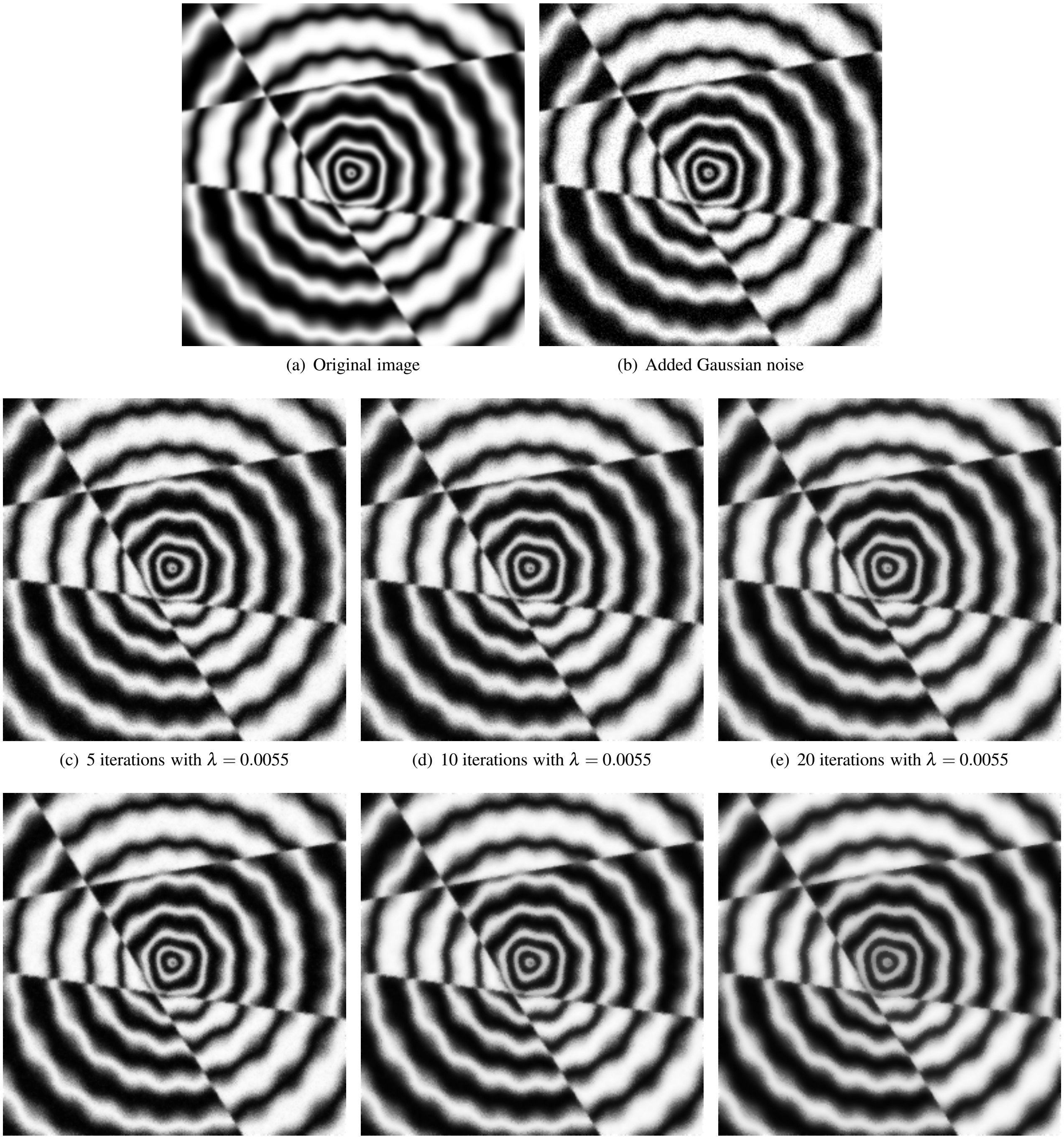
[4] Danny Barash. Bilateral filtering and anisotropic diffusion: Towards a unified viewpoint. In *In Third International Conference on ScaleSpace and Morphology*, pages 273–280, 2000.

[5] Peter Blomgren and Tony F. Chan. Color tv: Total variation methods for restoration of vector valued images. *IEEE Trans. Image Processing*, 7:304–309, 1996.

[6] V. De Castro, A.T. Smith, A.L. Beer, C. Leguen, N. Vayssière, Y. Héjja-Brichard, P. Audurier, B.R. Cottereau, and J.B. Durand. Connectivity of the Cingulate Sulcus Visual Area (CSv) in Macaque Monkeys. In *Cerebral Cortex*, volume 31, pages 1347–1364, 2020.

[7] Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. Bilateral filtering: Theory and applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1):1–73, 2009.

[8] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, November 1992.



(a) Original image (b) Added Gaussian noise

(c) 5 iterations with $\lambda = 0.0055$ (d) 10 iterations with $\lambda = 0.0055$ (e) 20 iterations with $\lambda = 0.0055$

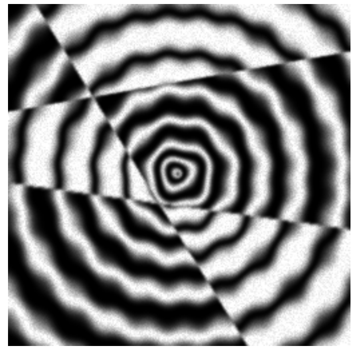
(f) 5 iterations with $\lambda = 0.02$ (g) 10 iterations with $\lambda = 0.02$ (h) 20 iterations with $\lambda = 0.02$

Figure 3. *Adding and removing Gaussian noise from the synthetic data. The original image is processed with 5, 10, and 20 iterations respectively.* λ *is set to* 0.0055 *and* 0.02.

[9] G. Sapiro and D. L. Ringach. Anisotropic diffusion of multivalued images with applications to color filtering. *IEEE Transactions on Image Processing*, 5(11):1582–1586, Nov 1996.

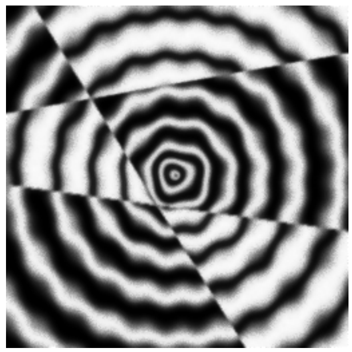
[10] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Iccv*, volume 98, page 2, 1998.

(a) Original image

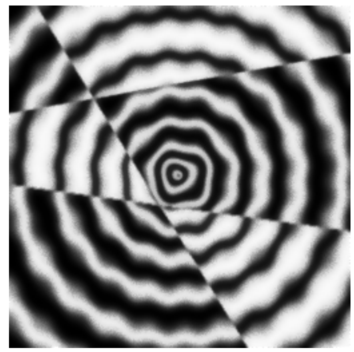


(b) Added speckle noise

(c) 5 iterations with $\lambda = 0.0055$

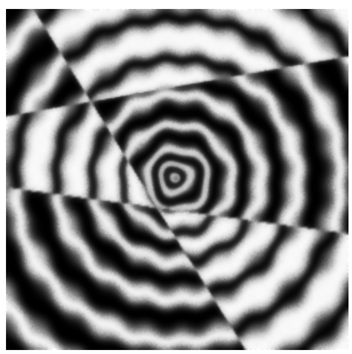


(d) 10 iterations with $\lambda = 0.0055$

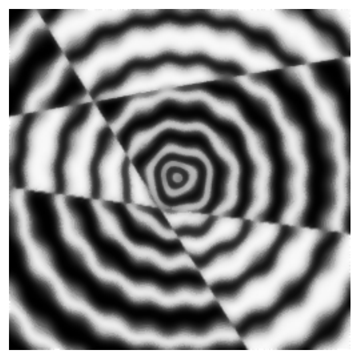


(e) 20 iterations with $\lambda = 0.0055$

(f) 5 iterations with $\lambda = 0.02$



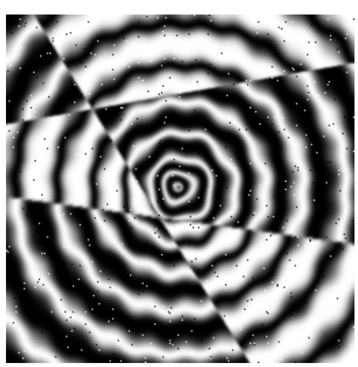
(g) 10 iterations with $\lambda = 0.02$



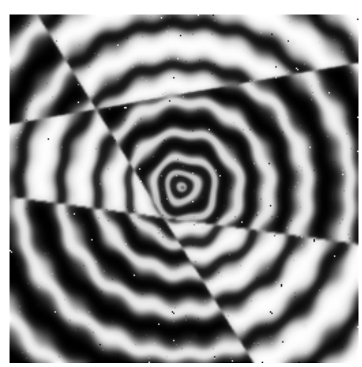
(h) 20 iterations with $\lambda = 0.02$

Figure 4. *Adding and removing Speckle noise from the example data. The original image is processed with 5, 10, and 20 iterations respectively.* λ *is set to* 0.0055 *and* 0.02.

(a) Original image

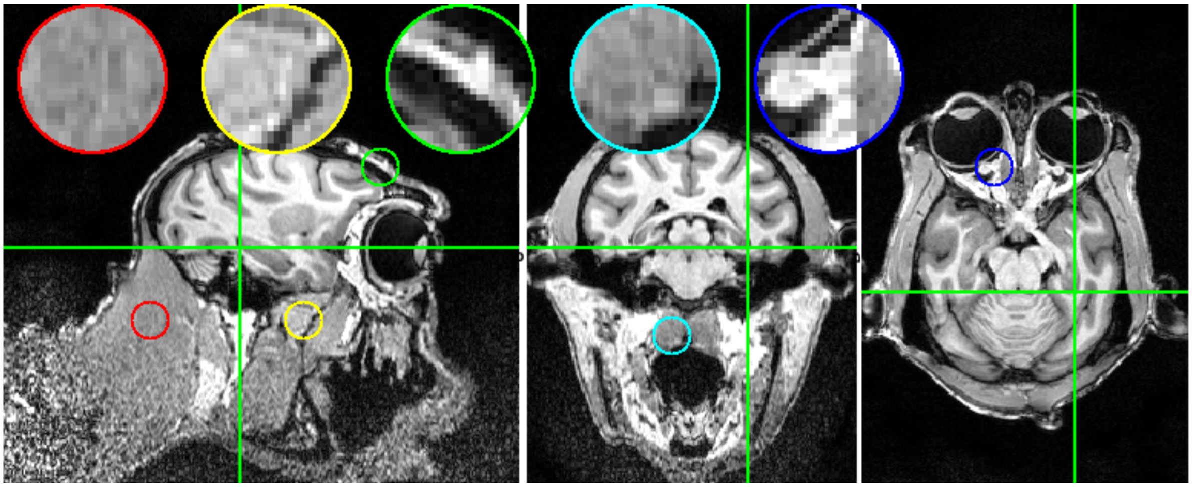


(b) Added Salt&Pepper noise

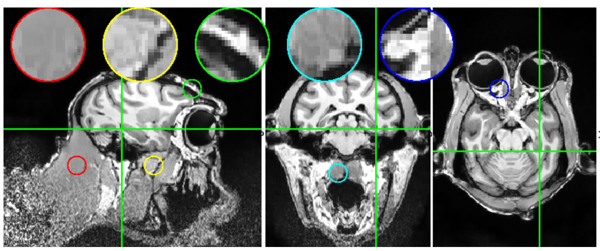


(c) Salt&Pepper 20 iterations

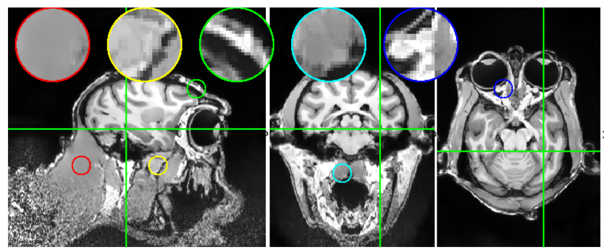
Figure 5. *Adding and removing Salt&Pepper noise from the synthetic data. The image is processed with 20 iterations respectively.* λ *is set to* 0.0055.



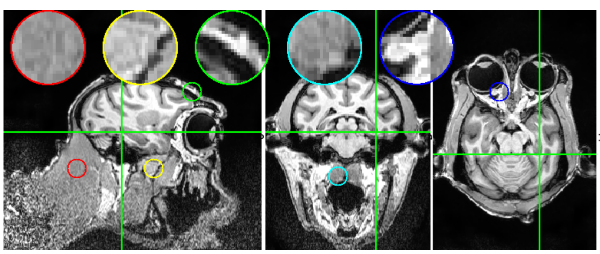
(a) Original MRI image



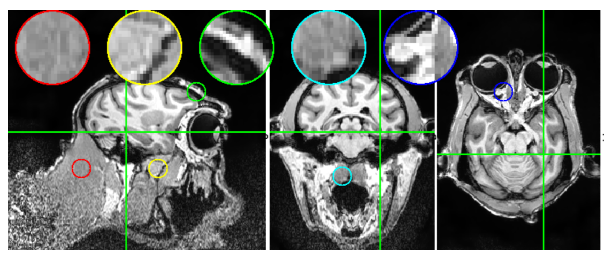
(b) $\lambda = 10^{-4}, n = 10$



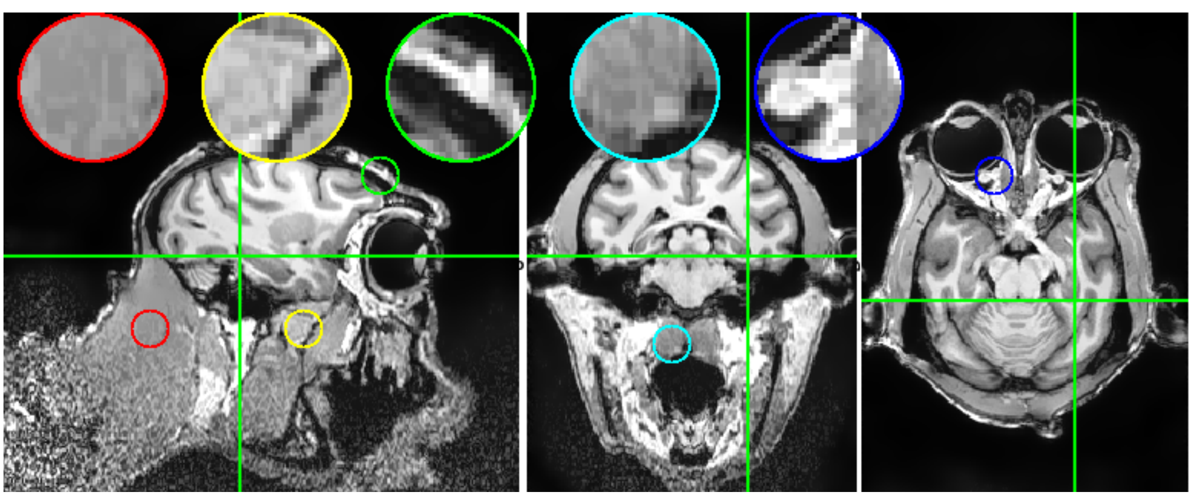
(c) $\lambda = 10^{-4}, n = 25$



(d) $\lambda = 10^{-5}, n = 10$



(e) $\lambda = 10^{-5}, n = 25$



(f) $\lambda = 10^{-5}, n = 100$

Figure 6. *The algorithm is applied on MRI-data of a Monkey. λ is set to $1.0e-4$ and $1.0e-5$. The algorithm ran with 10, 25 and 100 iterations.*