ECE 6560 Final Project

Noise Reduction

Jay Balar

1 Introduction

Noise in an image is the presence of random variations that are not found in the original scene [1]. The major concern with noise is that it is an unwanted and disturbing presence that diminishes the quality of an image. To address this issue, this project discusses and implements several different methods using PDEs to explore various ways to reduce noise in images.

2 Mathematical Formulation

Anisotropic Diffusion

A common method to reduce the levels of noise in an image is anisotropic diffusion. It focuses on edges, which can be considered sharp changes in image intensity [2]. Anisotropic diffusion is a method used to preserve edges while an image is being processed by impeding diffusion when an edge is reached. To do so, there is a function that limits image smoothing at an edge, referred to as the diffusion coefficient, $c(\mathbf{x})$. If $c(\mathbf{x})$ varies according to the local image gradient, then anisotropic diffusion is achieved. One basic model for an anisotropic model is [2]

$$\frac{\partial I_t(x)}{\partial t} = div\{c(\boldsymbol{x})\nabla I_t(x)\} \tag{1}$$

In this project, several different versions of anisotropic diffusion are tested, using different diffusion coefficients and diffusion PDEs to compare the level to which noise in images was reduced.

3 PDEs Utilized

In this section, the derivations of all the PDEs and diffusion coefficients chosen for this project will be briefly discussed.

3-1 PDE-2

One of the anisotropic models tested in this project is defined below [3], which is based on the diffusion model.

$$I_t = div(c(x,y,t)\nabla I) = c(x,y,t)\Delta I + \nabla c \cdot \nabla I \tag{2}$$

The value of the diffusion coefficient is defined as [3]

$$c(x,y,t) = \frac{1}{\sqrt{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}} \tag{3}$$

where K is just a constant. Substituting into equation 2 results in and assuming the diffusion coefficient is constant at time *t*, the following is obtained [4]

$$I_t = \alpha\left(I_{xx} + I_{yy}\right) \tag{4}$$

where α is equal to eq. 3.

3-2 PDE-4

The second anisotropic model tested was a fourth order PDE. This model is by You and Kaveh and below follows a summarized derivation [5]. They defined the energy functional as

$$E(u) = \iint f(|\nabla^2 u|)dxdy$$

The Euler equation was then derived as

$$\nabla^2[f'(|\nabla^2 u|sign(\nabla^2 u)] = 0$$

where

$$sign(s)\begin{cases} -1 & s < 0 \\ 0 & s = 0 \\ 1 & s > 0 \end{cases}$$

and

$$f'(s) = \frac{\partial f}{\partial s}$$

The Euler equation can be rewritten as

$$\nabla^2[f'(|\nabla^2 u|)\frac{\nabla^2 u}{|\nabla^2 u|}] = 0, \qquad \left.\frac{\nabla^2 u}{|\nabla^2 u|}\right|_{\nabla^2 u=0} = 0$$

Given the equation [5]

$$c(s) = \frac{f'(s)}{s}$$

then

$$\nabla^2[c(|\nabla^2 u|)|\nabla^2 u|] = 0$$

This equation can be solved through the following gradient descent procedure [5]

$$\frac{\partial u}{\partial t} = -\nabla^2[c(|\nabla^2 u|)|\nabla^2 u|$$

Full derivation details can be found in the referenced paper [5]. The implementation used for this project is [4]

$$I_t = -\alpha\left(I_{xxxx} + I_{yyyy}\right) \tag{5}$$

The value of the diffusion coefficient was chosen as [4]

$$\alpha = \frac{1}{\sqrt{1 + \left(\frac{|\nabla^2 I|}{K}\right)^2}} \tag{6}$$

3-3 Fidelity Term

The next method that was analyzed was obtained using an energy functional from the class exam as shown below

$$E(I) = \iint \frac{1-\lambda}{2}\|\nabla I\|^2 + \frac{\lambda}{2}(I - I_0)^2$$

where I_0 is the original image and λ represents a constant value from 0 to 1 which weighs fidelity against smoothness. The Langrangian is defined as

$$L = \frac{1-\lambda}{2}\left({I_x}^2 + {I_y}^2\right) + \frac{\lambda}{2}(I - I_0)^2$$

The steps for deriving the Euler Lagrange equation are

$$L_I - \frac{\partial}{\partial x}L_{I_x} - \frac{\partial}{\partial y}L_{I_y} = 0$$

$$L_I = \lambda(I - I_0)$$

$$L_{I_x} = (1-\lambda)I_x$$

$$L_{I_y} = (1-\lambda)I_y$$

$$\frac{\partial}{\partial x}L_{I_x} = (1-\lambda)I_{xx}$$

$$\frac{\partial}{\partial y}L_{I_y} = (1-\lambda)I_{yy}$$

$$\lambda(I - I_0) - (1-\lambda)\Delta I = 0$$

$$I_t = -\nabla_I E = -\lambda(I - I_0) + (1-\lambda)\Delta I \tag{7}$$

3-4 Speckle-Removing Anisotropic Diffusion

The diffusion equation used for this model was taken from Yu and Acton [6] with the PDE

$$\begin{cases} \dfrac{\partial I(x,y;t)}{\partial t} = div[c(q)\nabla I(x,y;t)] \\ I(x,y;0) = I_0(x,y), (\dfrac{\partial I(x,y;t)}{\partial \vec{n}})\Big|_{\partial\Omega=0} \end{cases} \tag{8}$$

with

$$c(q) = \frac{1}{1 + [q^2(x,y;t) - {q_0}^2(t)]/[{q_0}^2(t)(1 + {q_0}^2(t)]} \; or$$

$$c(q) = \exp\{-\frac{[q^2(x,y;t) - {q_0}^2(t)]}{[{q_0}^2(t)(1 + {q_0}^2(t))]}\}$$

The value $q(x,y;t)$ is defined as the instantaneous coefficient of variation and is determined by

$$q(x,y;t) = \sqrt{\frac{\left(\frac{1}{2}\right)\left(\frac{|\nabla I|}{I}\right)^2 - \left(\frac{1}{4}\right)^2 (\frac{\nabla^2 I}{I})^2}{[1 + (\frac{1}{4})(\frac{\nabla^2 I}{I})]^2}}$$

The function $q_0(t)$ is the speckle scale function and defined as

$$q_0(t) = \frac{\sqrt{var[z(t)]}}{\overline{z(t)}}$$

where $z(t)$ represents the intensity over the area *t* [6].

3-5 Projected Mean Curvature Diffusion PDE

In this model, a 3D graph *s* can be defined as [2]

$$s(\boldsymbol{x}) = s(x,y) = [x, y, I(x,y)], \boldsymbol{x} = (x,y)$$

Applying mean curvature results in the PDE [2]

$$\frac{\partial s(\boldsymbol{x})}{\partial t} = h(\boldsymbol{x})n(\boldsymbol{x})$$

The function ***h(x)*** is the mean curvature and is determined as [2]

$$h(x,y) = \frac{\frac{\partial^2 I(x,y)}{\partial x^2}\left[1 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2\right] - 2\left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right)\left(\frac{\partial^2 I(x,y)}{\partial x \partial y}\right) + \frac{\partial^2 I(x,y)}{\partial y^2}\left[1 + \left(\frac{\partial I(x,y)}{\partial x}\right)^2\right]}{2\{1 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2\}^{3/2}}$$

The value ***n*(x)** is the unit normal to the surface and defined as [2]

$$n(x,y) = \frac{[\frac{-\partial I(x,y)}{\partial x}, \frac{-\partial I(x,y)}{\partial x}, 1]}{\sqrt{1 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2 + \left(\frac{\partial I(x,y)}{\partial y}\right)^2}}$$

Projecting the mean curvature diffusion PDE results in [2]

$$\frac{\partial s(\boldsymbol{x})}{\partial t} = \{[h(\boldsymbol{x})n(\boldsymbol{x})] \cdot z\}z$$

where z is the unit vector in the vertical direction of the graph *s*. Then, the PDE for updating the image becomes [2]

$$\frac{\partial I(\boldsymbol{x})}{dt} = \frac{\Delta I(x,y) + k^2[\left(\frac{\partial I(x,y)}{\partial x}\right)^2 \left(\frac{\partial^2 I(x,y)}{\partial y^2}\right) - 2\left(\frac{\partial I(x,y)}{\partial x}\right)\left(\frac{\partial I(x,y)}{\partial y}\right)\left(\frac{\partial^2 I(x,y)}{\partial x \partial y}\right) + \left(\frac{\partial I(x,y)}{\partial y}\right)^2 \left(\frac{\partial^2 I(x,y)}{\partial x^2}\right)]}{\{1 + k^2[\nabla I(x,y)]^2\}^2} \quad (9)$$

3-6 Multigrid

The diffusion PDE can be given by [7]

$$\frac{\partial I(\boldsymbol{x})}{\partial t} = div(c\nabla I)$$

Following the definition provided by Acton [7], the implementation is

$$[I(\boldsymbol{x})]_{t+1} = [I(\boldsymbol{x}) + \frac{1}{\Omega}\sum_{d=1}^{\Omega} c_d(\boldsymbol{x})\nabla I_d(\boldsymbol{x})]]_t$$

The value Ω is the number of directions diffusion is calculated and the subscripts *d*, indicate the direction. In two dimensions, the equation can be written as [7]

$$[I(x,y)]_{t+1} = \{I(x,y) + \frac{1}{\Omega}[c_N(x,y)\nabla I_N(x,y) + c_S(x,y)\nabla I_S(x,y) + c_E(x,y)\nabla I_E(x,y) + c_W(x,y)\nabla I_W(x,y)]\}_t \quad (10)$$

where the subscripts represent the four cardinal directions and Ω would generally be four.

4 Discrete Implementation

For all the implementations described above, the discrete models used a FTCS scheme. This was utilized as the central derivative approximation is generally the most accurate comparatively to forwards or backwards. The problem of edge conditions arose, however, as a central difference implementation relies on the values ahead of and behind the current pixel. So, if the pixel is on the outer boundary, then there will be no previous or following pixel in at least one direction. Due to this, all the implementations that follow will start from the second index of both the row and column of the image matrix and end before the last index of both column and row. While this does neglect noise at the edges, for the images considered in this project which consisted of several hundred rows and columns, any negative effect will be minimal.

Also worth noting is that although the discrete implementations shown below are all for two dimensional images, experiments were conducted using color images. However, the implementations all still apply. Each image would be of size *m*n*o*. By setting *o* as a constant, each implementation can be applied on a two-dimensional matrix. Using this method for a color image, all that needs to be done is to repeat the implementation from 1 to *o* for each *m*n* matrix. In general, all the PDEs discussed will follow the same algorithm as shown below.

Generic Algorithm for PDEs

---

1) Read in image
2) Initialize $\Delta x, \Delta y, \Delta t$ and set the number of repetitions
3) While num=1: num_repetitions
    While k = 1: *o* (size of the third matrix dimension)
        While j = 2: m-1 (number of rows ignoring edges)
            While i = 2: n-1 (number of columns ignoring edges)
                Calculate diffusion coefficients and update the PDE
            End
        End
    End
End

Next, all the PDEs derived in the previous section need to be discretized. This process is discussed for each PDE in the following sections.

4-1 PDE-2

For this PDE, the second order central difference formulas are needed, which are

$$I_{xx} = \frac{I(x+\Delta x, y, t) - 2I(x, y, t) + I(x-\Delta x, y, t)}{\Delta x^2} \tag{11}$$

$$I_{yy} = \frac{I(x, y+\Delta y, t) - 2I(x, y, t) + I(x, y-\Delta y, t)}{\Delta y^2} \tag{12}$$

with an error term of $O(\Delta x^2)$. The gradient can be defined as

$$\nabla I = [\frac{I(x+\Delta x, y, t) - I(x-\Delta x, y, t)}{2\Delta x}, \frac{I(x, y+\Delta y, t) - I(x, y-\Delta y, t)}{2\Delta y}] \tag{13}$$

Substituting into the equation 4, the result is

$$I(x, y, t+\Delta t) = I(x, y, t) + \frac{\Delta t}{\Delta x^2}\alpha\big(I(x+\Delta x, y, t) - 2I(x, y, t) + I(x-\Delta x, y, t)\big) + \frac{\Delta t}{\Delta y^2}\alpha(I(x, y+\Delta y, t) - 2I(x, y, t) + I(x, y-\Delta y, t)) \tag{14}$$

where α is as defined in equation 3.

4-2 PDE-4

This PDE requires the fourth order central difference equations, which are defined as

$$Ixxxx = \frac{I(x-2\Delta x,y,t) - 4I(x-\Delta x,y,t) + 6I(x,y,t) - 4I(x+\Delta x,y,t) + I(x+2\Delta x,y,t)}{(\Delta x)^4} \tag{15}$$

$$Iyyyy = \frac{I(x,y-2\Delta y,t) - 4I(x,y-\Delta y,t) + 6I(x,y,t) - 4I(x,y+\Delta y,t) + I(x,y+2\Delta y,t)}{(\Delta y)^4} \tag{16}$$

Substituting in from equation 5, the result is

$$\begin{aligned} I(x,y,t+\Delta t) = I(x,y,t) &- \frac{\Delta t}{\Delta x^4}\alpha\big(I(x-2\Delta x,y,t) - 4I(x-\Delta x,y,t) + 6I(x,y,t) - 4I(x+\Delta x,y,t) + I(x+2\Delta x,y,t)\big) \\ &- \frac{\Delta t}{\Delta y^4}\alpha(I(x,y-2\Delta y,t) - 4I(x,y-\Delta y,t) + 6I(x,y,t) - 4I(x,y+\Delta y,t) + I(x,y+2\Delta y,t)) \end{aligned} \tag{17}$$

$$\nabla^2 I = I_{xx} + I_{yy} \tag{18}$$

where α is defined in equation 6 and I_{xx} and I_{yy} are defined in equations 11 and 12.

4-3 Fidelity Term

By defining the forward difference in time as

$$I_t = \frac{I(x,y,t+\Delta t) - I(x,y,t)}{\Delta t}$$

and using equation 7, the following can be obtained

$$\frac{I(x,y,t+\Delta t) - I(x,y,t)}{\Delta t} = I(x,y,t) - \lambda\Delta t\big(I(x,y,t) - I_0(x,y)\big) + (1-\lambda)\frac{\Delta t}{\Delta x^2}I_{xx} + (1-\lambda)\frac{\Delta t}{\Delta y^2}I_{yy}$$

which can be rewritten using equations 11 and 12 as

$$\begin{aligned} I(x,y,t+\Delta t) = I(x,y,t) - \lambda\Delta t\big(I(x,y,t) - I_0(x,y)\big) + (1-\lambda)\frac{\Delta t}{\Delta x^2}\big(I(x+\Delta x,y,t) - 2I(x,y,t) + I(x-\Delta x,y,t)\big) + \\ (1-\lambda)\frac{\Delta t}{\Delta y^2}(I(x,y+\Delta y,t) - 2I(x,y,t) + I(x,y-\Delta y,t)) \end{aligned} \tag{19}$$

The value for λ was chosen to be 0.5.

4-4 Speckle-Removing Anisotropic Diffusion

Following the method implemented by Yu and Acton [6], value for $q_0(t)$ can be approximated as

$$q_0(t) \approx q_0 \exp\left[-\rho t\right]$$

where ρ is a constant and q_0 is the speckle coefficient of variation. With this, Yu and Acton defined the discrete isotropic diffusion update as [6]

$$I(x,y,t+\Delta t) = I(x,y,t) + \frac{\Delta t}{4}(I(x+\Delta x,y,t) + I(x-\Delta x,y,t) + I(x,y+\Delta y,t) + I(x,y-\Delta y,t) - 4I(x,y,t)) \quad (20)$$

4-5 Projected Mean Curvature Diffusion PDE

The diffusion PDE defined in equation 9 can be rewritten as

$$I_t = \frac{\Delta I + k^2(I_x{}^2 I_{yy} - 2I_x I_y I_{xy} + I_y{}^2 I_{xx})}{(1 + k^2\|\nabla I\|^2)^2}$$

Discretizing this equation results in the following

$$\begin{aligned} I(x,y,t+\Delta t) = I(x,y,t) + \frac{\Delta t}{\left(1 + k^2(\|\nabla I\|)\right)^2}\{\Delta I + k^2(&\frac{1}{\Delta x^2}\left(\frac{I(x+\Delta x,y,t) - I(x-\Delta x,y,t)}{2}\right)^2 \\ &* \frac{1}{\Delta y^2}\left(I(x,y+\Delta y,t) - 2I(x,y,t) + I(x,y-\Delta y,t)\right) - 2\left(\frac{1}{\Delta x}\right)\left(\frac{I(x+\Delta x,y,t) - I(x-\Delta x,y,t)}{2}\right) \\ &* \left(\frac{1}{\Delta y}\right)\left(\frac{I(x,y+\Delta y,t) - I(x,y-\Delta y,t)}{2}\right) * (\frac{1}{\Delta x \Delta y})(\frac{1}{4}(I(x-\Delta x,y-\Delta y,t) + I(x+\Delta x,y+\Delta y,t) \\ &- I(x+\Delta x,y-\Delta y,t) - I(x-\Delta x,y+\Delta y,t) + \frac{1}{\Delta y^2}\left(\frac{I(x,y+\Delta y,t) - I(x,y-\Delta y,t)}{2}\right)^2 \\ &* \frac{1}{\Delta x^2}\left(I(x+\Delta x,y,t) - 2I(x,y,t) + I(x-\Delta x,y,t)\right)\} \end{aligned} \quad (21)$$

$$\Delta I = I_{xx} + I_{yy} \quad (22)$$

where I_{xx} and I_{yy} are defined by equations 11 and 12.

4-6 Multigrid

The directional derivatives mentioned in section 3-6 can be discretized as

$$\nabla I_N = \frac{I(x,y+\Delta y,t) - I(x,y,t)}{\Delta y} \quad (23)$$

$$\nabla I_S = \frac{I(x,y-\Delta y,t) - I(x,y,t)}{\Delta y} \quad (24)$$

$$\nabla I_E = \frac{I(x+\Delta x,y,t) - I(x,y,t)}{\Delta x} \quad (25)$$

$$\nabla I_W = \frac{I(x-\Delta x,y,t) - I(x,y,t)}{\Delta x} \quad (26)$$

Then, the diffusion coefficient can be defined as shown below [7],

$$c_N = \exp\left(-\left(\frac{\nabla I_N}{K}\right)^2\right) \quad (27)$$

$$c_S = \exp\left(-\left(\frac{\nabla I_S}{K}\right)^2\right) \quad (28)$$

$$c_E = \exp\left(-\left(\frac{\nabla I_E}{K}\right)^2\right) \tag{29}$$

$$c_W = \exp\left(-\left(\frac{\nabla I_W}{K}\right)^2\right) \tag{30}$$

Substituting all this into the update equation results in

$$\begin{aligned} I(x,y,t+\Delta t) = I(x,y,t) \\ + \frac{1}{\Omega}[c_N(x,y,t)\nabla I_N(x,y,t) + c_S(x,y,t)\nabla I_S(x,y,t) + c_E(x,y,t)\nabla I_E(x,y,t) \\ + c_W(x,y,t)\nabla I_W(x,y,t)], \qquad \Omega = 4 \end{aligned} \tag{31}$$

5 Experimental Results

Experiments were conducted using the set of images shown below. Although Image 1 can be considered greyscale, it was read as an RGB image for this project.

Image 1 [8]

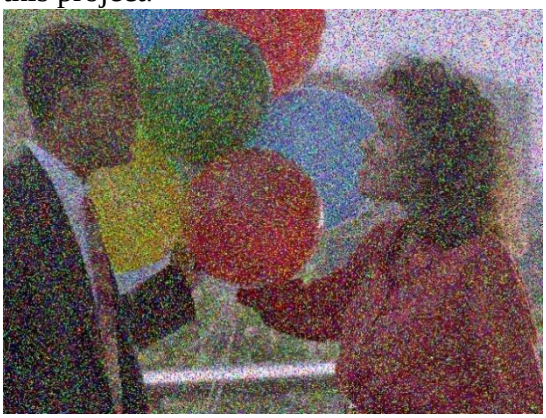

Image 2 [9]

Gaussian noise was added to Image 1, using the default settings for the Matlab function imnoise, to test the noise removal ability of all the PDEs, resulting in the image shown below. All the code used can be found https://github.com/jbalar3/ECE-6560-Final-Project.

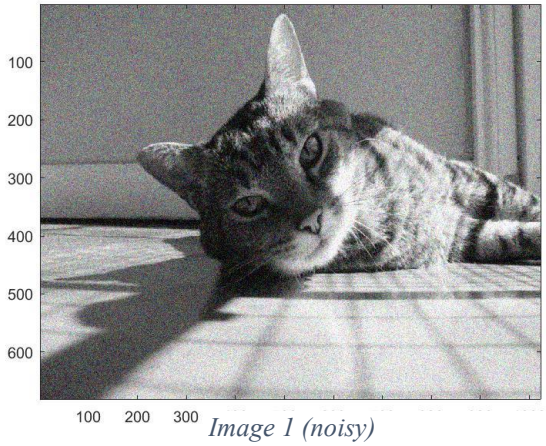

Image 1 (noisy)

The implementations were applied for Image 1 (noisy) and Image 2, and the results are shown next.

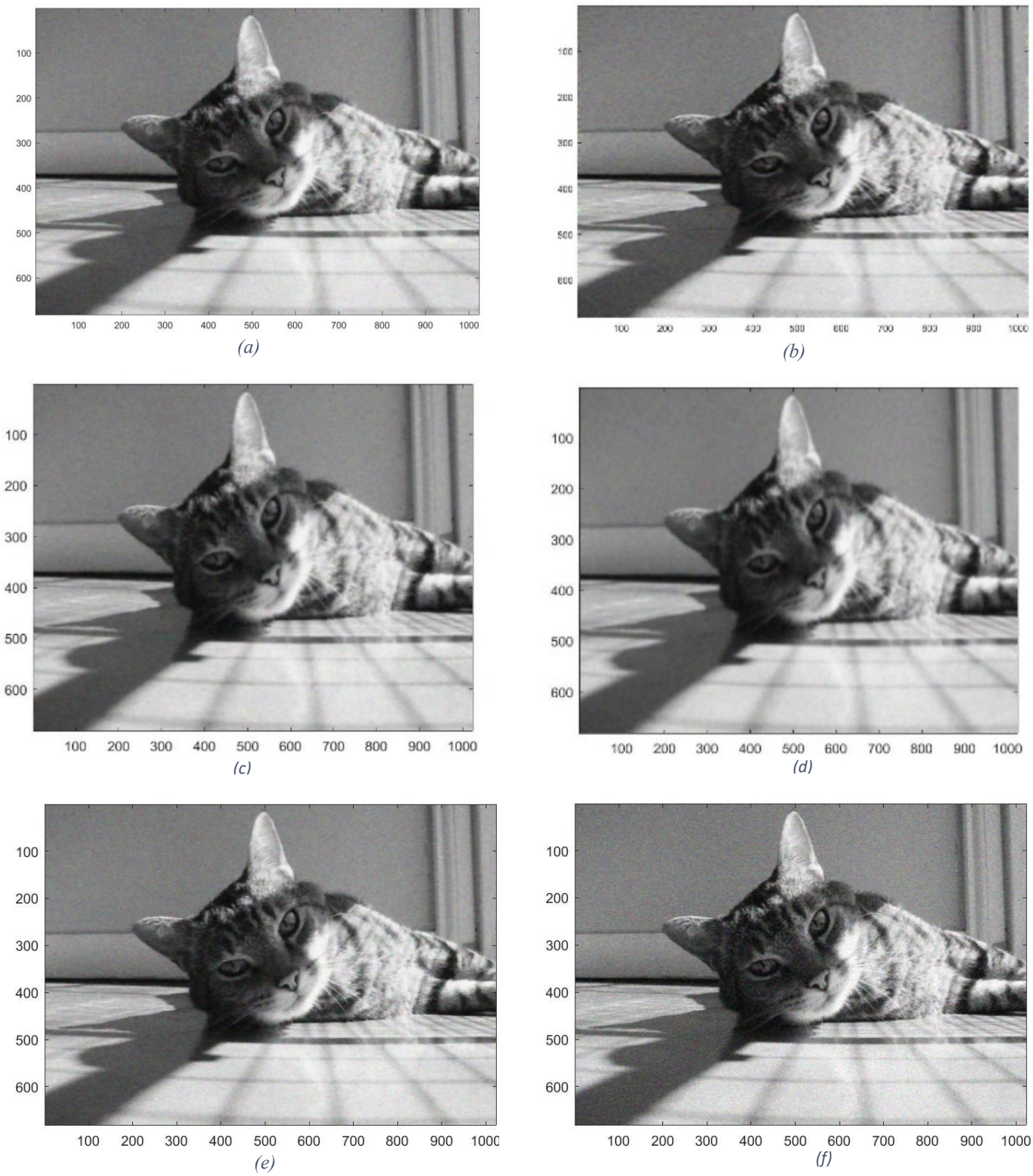

Figure 1. (a) Five iterations of Multigrid (b) 120 iterations of PDE-4 (c) 8 iterations of the projected mean curvature diffusion PDE (d) 100 iterations of the speckle-removing PDE (e) 20 iterations of PDE-2 (f) 20 iterations of Fidelity

Figure 2. (a) 180 iterations of PDE-4 (b) 40 iterations PDE-2 (c) 20 iterations of the Fidelity (d) 100 iterations of the speckle-removing PDE (e) 50 iterations of the projected mean curvature Diffusion PDE (f) 15 iterations of Multigrid

Visually inspecting the images shown in Figure 1 and 2, there does seem to be a noticeable reduction in the noise levels present. To confirm this, plots of the image intensity were recorded. These profiles were chosen as a random segment of the image. The intensity of Image 2 along the chosen profile is shown below

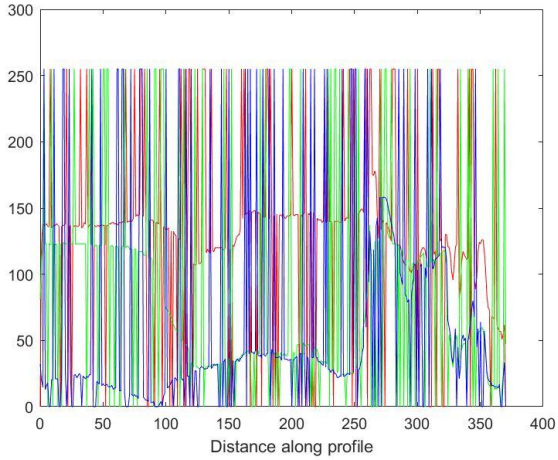

Figure 3. Intensity plot of Image 2

Next, the intensity plots for each implementation using Image 2 were obtained. They are shown below, where the profile for each was also randomly selected.

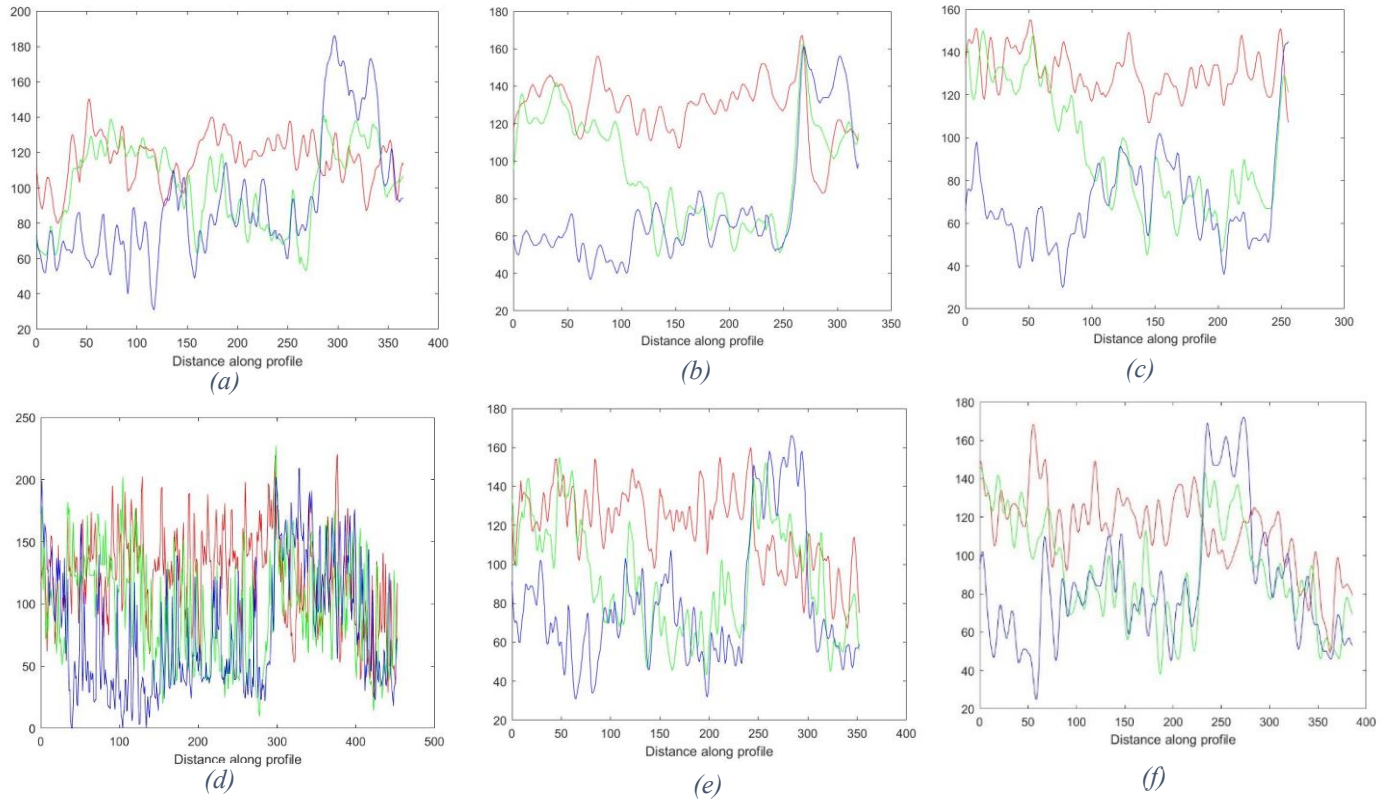

Figure 4. Plots for (a) Multigrid (b) Projected Mean Curvature Diffusion PDE (c) Speckle (d) Fidelity (e) PDE-2 (f) PDE-4

Compared to the profile intensity for the original image, all the implementations do have substantially less variation in their intensity profiles. It seems that almost all the implementations tested tend to have much smoother profile compared to the original image, except for the Fidelity model. This is due to Image 2 having a great deal of noise, making it a very poor reference photo. Since the Fidelity model attempts to remain close to the initial image, the result would filter out less noise compared to the other models.

Another metric considered for evaluating the noise levels of the filtered images was the peak-signal to noise ratio (PSNR). PSNR is the ratio between the maximum value of a signal and the value of the noise that is affecting the image [10].

The value for the PSNR was calculated using the formula below

$$MSE = \frac{1}{m * n * 3} \sum_{k=1}^{3} \sum_{i=2}^{m-1} \sum_{j=2}^{n-1} [I(i,j,k) - N(i,j,k)]^2$$

$$PSNR = 20log_{10}(\frac{MAX_I}{\sqrt{MSE}})$$

where MSE is the mean square error, where *I* is the noise-free reference image and *N* is the noisy approximation. MAX_I is the maximum value of a pixel in image *I*. The index for the row and column summation terms starts at two and ends before the final index respectively for the columns and rows as the outer boundaries were ignored for each implementation. The PSNR values were calculated using Image 1 as a reference.

Table 1: PSNR Values for the PDEs

PDE	PSNR	Iterations
PDE-4	30.4018	120
PDE-2	31.2977	20
Fidelity	29.0806	20
Speckle	30.2361	100
Projected Mean Curvature Diffusion	30.8639	8
Multigrid	30.7791	5

Overall, the PSNR values are very similar for each implementation with PDE-2 performing the best and the Fidelity model performing the worst.

6 Discussion

From the experiments conducted, it is difficult to conclude whether any one implementation is far superior to the rest. When visually comparing the results, there seemed to be little difference in quality amongst all the implementations, although it is a subjective judgement. And although the Fidelity model did perform the worst in terms of its measured PSNR and has the most variation in its intensity plot, it still holds significant value if prioritizing the fidelity of an image is paramount.

One of the major challenges of this project was determining the value for the constant term k, which was used in several of the implementations discussed. This value was used to scale the image gradient, which would vary depending on the size of the image. So, if k is too small, the coefficient of variation will approach zero, whereas if k is too large, then it will approach one. Due to this, it was important to be aware of the range of the image gradient to then choose an appropriate value for k to ensure that the coefficient of diffusion varies as a function of the gradient.

Another challenge was choosing an appropriate time step as well as the ideal number of iterations. Although the Courant-Friedrichs-Lewy (CFL) conditions were not explicitly calculated, stability requirements obviously needed to be met, so time steps were experimentally determined. The challenge in doing so, however, was to ensure that the time step was large enough to limit the number of iterations needed, but also small enough to not violate the stability criterion.

The number of iterations was also important to determine as, although noise would continue to diminish, the image would start to blur and lose finer details as iterations increased. The iteration number was determined through experimentation as well and differed greatly between implementations.

Overall, the implementations discussed did work as intended as noise could be diminished through all the methods considered. However, for all the PDEs outside of the fidelity model, finer details would be lost as the number of iterations increased. If the goal is to obtain a noise free image that resembles the original, then the PDE methods discussed should be paired with a PDE meant for image sharpening to ensure the highest quality image. Improvements that could be made for this project would be to calculate the CFL conditions to determine the ideal time step as well as being more selective when choosing the value of the constant term k. Edge conditions can also be taken into account by pairing the central difference method with a forward and backwards difference to ensure that all boundaries are accounted for.

7 References

[1] “Image Noise - Image Engineering,” *www.image-engineering.de*. https://www.image-engineering.de/library/image-quality/factors/1080-noise

[2] Acton, Scott. (2009). Diffusion Partial Differential Equations for Edge Detection. 10.1016/B978-0-12-374457-9.00020-2.

[3] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629-639, July 1990, doi: 10.1109/34.56205.

[4] S. A. Halim, A. Ibrahim, and Y. H. P. Manurung, 'The performance of PDE-based image denoising on radiographic images', *AIP Conference Proceedings*, vol. 1750, no. 1, 06 2016.

[5] Y. . -L. You and M. Kaveh, "Fourth-order partial differential equations for noise removal," in IEEE Transactions on Image Processing, vol. 9, no. 10, pp. 1723-1730, Oct. 2000, doi: 10.1109/83.869184.

[6] Yongjian Yu and S. T. Acton, "Speckle reducing anisotropic diffusion," in IEEE Transactions on Image Processing, vol. 11, no. 11, pp. 1260-1270, Nov. 2002, doi: 10.1109/TIP.2002.804276.

[7] S. T. Acton, "Multigrid anisotropic diffusion," in IEEE Transactions on Image Processing, vol. 7, no. 3, pp. 280-291, March 1998, doi: 10.1109/83.661178.

[8] K. Norton, "What is Grayscale?," *Webopedia*, Aug. 13, 2021. https://www.webopedia.com/definitions/grayscale/ (accessed Apr. 24, 2023).

[9] "image_denoise," *people.sc.fsu.edu*. https://people.sc.fsu.edu/~jburkardt/cpp_src/image_denoise/image_denoise.html (accessed Apr. 24, 2023).

[10] "Peak Signal-to-Noise Ratio as an Image Quality Metric," *www.ni.com*. https://www.ni.com/en-us/shop/data-acquisition-and-control/add-ons-for-data-acquisition-and-control/what-is-vision-development-module/peak-signal-to-noise-ratio-as-an-image-quality-metric.html (accessed Apr. 24, 2023).