Department of Computer Science
CS663 - Fundamentals of Image Processing

Report : Vector-Valued Image Regularization with PDEs

Prepared by: Desai Sai Pranav(210050043)
Pragallapati Venkata Ratna Sai Kumar(210050120)
Terli Tulsiram(210050157)

Date: November 27, 2023

Abstract of ideas we aimed to implement

Selected Paper Vector-Valued Image Regularization with PDEs: A Common Framework for Different Applications
Analyzing Functional minimization , Divergence expressions, oriented laplacians ,relating them to one another and understanding the geometric significance . After coming to a unified expression , modifying it further to a new vector-valued regularization PDE that follows desired local geometric properties Numerical implementation of the PDE so far without the need for calculating second derivatives by translating them to gaussian kernels and convolving our image with it . Experimenting these kernels to solve real life problems like
– Color Image Restoration of digital photograph with real noise, due to the bad lighting conditions
– Color Image inpainting of images covered with unwanted texts in a structure preserving way
– Color Image Magnification by linear interpolation of a small image, and applying our PDE and retrieving nonlinear magnified images without jagging or bloc effects

Conclusions from paper implemented

The final equations which we implemented in our code are

$$I_{i(t)} = I_{i_{(t=0)}} * G^{(\mathbf{T},t)}$$

where $*$ stands for the convolution with the oriented Gaussian kernel

$$G^{(\mathbf{T},t)}(\mathbf{x}) = \frac{1}{4\pi t} \exp\left(-\frac{\mathbf{x}^T\mathbf{T}^{-1}\mathbf{x}}{4t}\right) \quad \text{with} \quad \mathbf{x} = (x \quad y)^T$$

where $\mathbf{T}$ is the tensor field defined pointwise as

$$\mathbf{T} = \frac{1}{\sqrt{1+\lambda_+^* + \lambda_-^*}}\theta_-^*\theta_-^{*T} + \frac{1}{1+\lambda_+^* + \lambda_-^*}\theta_+^*\theta_+^*$$

where $\lambda_\pm^*$ and $\theta_\pm^*$ are defined to be the spectral elements of $G_\sigma = G * G_{\sigma'}$, a Gaussian smoothed version of the structure tensor G, allowing us to retrieve a more coherent vector geometry giving a better approximation of the vector discontinuity directions.

These are the solutions of equations, (solved in the paper)

$$\frac{\partial I_i}{\partial t} = c_1 I_{i\epsilon\varepsilon} + c_2 I_{im} = \text{trace}\left(\mathbf{T}H_i\right)$$

$$\frac{\partial I_i}{\partial t} = \text{trace}\left(\mathbf{T}\mathbf{H}_i\right) \quad (i = 1..n)$$

where H_i is the Hessian matrix of the vector component I_i and T is the 2x2 tensor defined by:

$$\mathbf{T} = c_1\xi\xi^T + c_2\eta\eta^T$$

characterized by its two eigenvalues c1,c2 and its two corresponding eigenvectors η,ξ

We are basically moving through each point , calculating the structure tensor there and then smooth it and transform the eigen values to smooth in different amounts in different directions, then apply gaussian kernel to the selected neighbourhood.

The functions we wrote which transform the data take the following as arguments

- filepath : input image path
- num_iterations
- t : equivalent for time in the above PDE
- neighbourhood : The number of pixels which affect this pixel

Point to be noted : In all functions, we don't apply this procedure to each and every pixel, for example in experiment where 50% pixels were erased, then there is no need to apply to the pixels which weren't affected , similarly to image where we are removing with the help of a mask, there is no need to apply to pixels which are already correct . For noise removal we need to apply to all pixels because we dont know where noise is , but there maybe some effect of doing so to normal pixels in short iteration count.

1 Observations and Conclusions

1. In the experiment of erasing pixels, when the percentage of pixels erased is lot greater than 50 , the results won't be very satisifactory

2. Though an experiment where erasing 50 percent pixels and then restoring using the above process worked well, image down sampling and then expanding by nearest neighbourhood estimation and then applying the above equations didn't bring the expected results because loss of data in some areas is unreconstructable .

3. Image In-painting takes lot of time to run compared to run, as the number of iterations needed to bring a visible change itself is very large because some images have very large thick masks to remove and the change to penetrate to the first layer is very feeble in one iteration , hence requires very high number of iterations and thus more time to run

4. Although the results were inserted above, they were not run with very high neighbourhood windows and iterations because of the limitations of time. So, better results can be produced than shown above.

1.1 Image denoising when there is not proper light

Figure 1.1: above is a noisy image

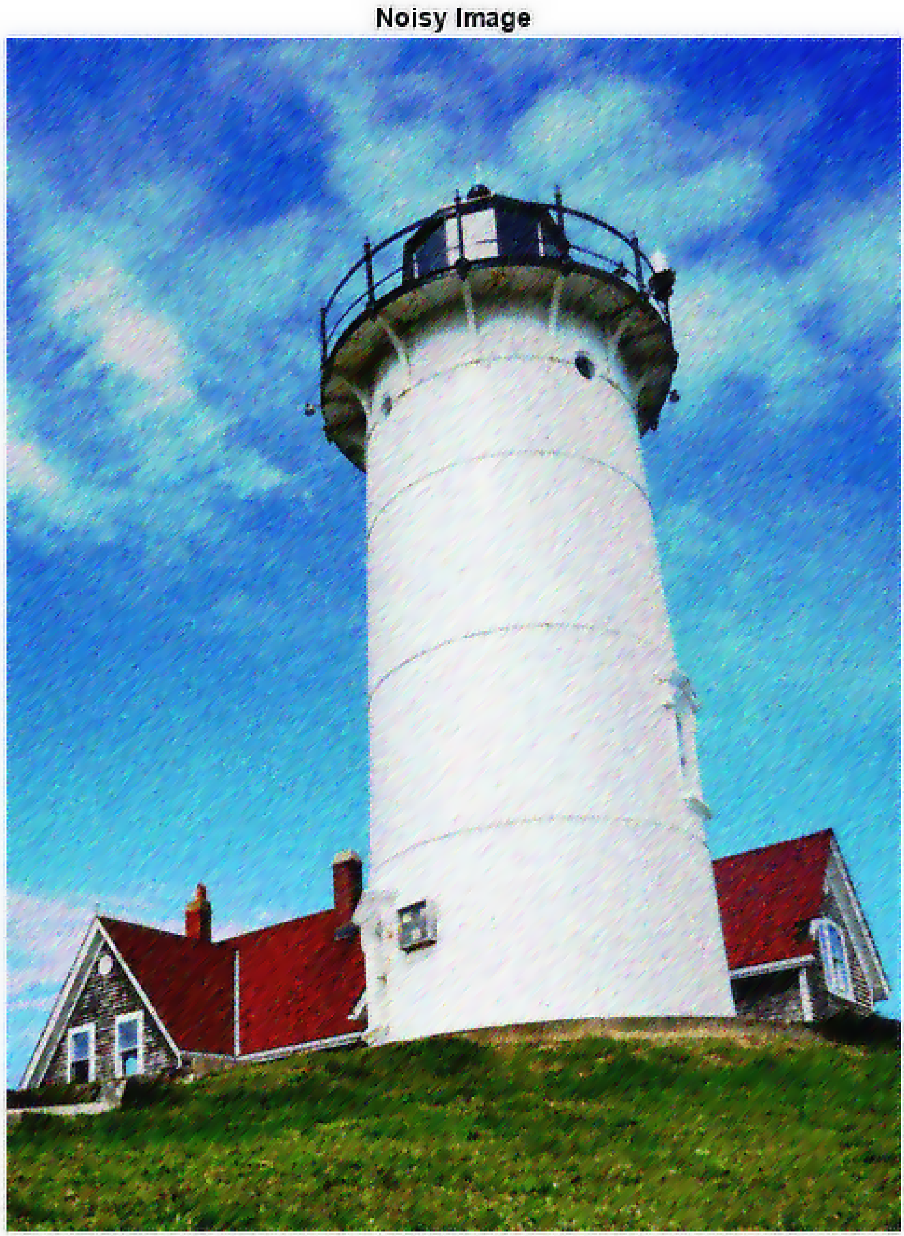

Figure 1.2: After applying the PDE with neighbouhood 7x7 , we can see lot of improvement

1.2 Filling missing pixels

Figure 1.3: Initial image of tree , After removing 50 percent pixels

Figure 1.4: Image after applying the PDE , we can clearly see the white pixels are retained , black turned white outside the tree and inside the tree , the colours are well reconstructed

1.3 Image Restoration

Figure 1.5: Initial image of bheem , After downsampling it by 3 times and zooming it again

Figure 1.6: Image after applying the PDE , we can clearly see the white pixels are retained , black turned white outside the tree and inside the tree , the colours are well reconstructed. We can observe the reconstruction with 5 epochs is not very impressive because lots of data is already lost like the lines and borders when downsampling

1.4 Removing obstructing objects

Figure 1.7: above parrot inside a cage, below image after applying the PDE with neighbourhood 5x5 with 21 iterations . we can observe that, still there is only a small blur despite so many iterations