CS663 - Project Report

Vector-Valued Image Regularization with PDEs

Group Members : Maloth Maheer (180050054)
Anish Deshpande (180100013)
Deep Satra (180040030)
Ayush Garg (180050016)

Contents

1 Abstract

We focused on techniques for vector-valued image regularization, based on variational methods and PDEs. PDE's help studying situations that arise in fluid mechanics, heat transfer, electromagnetism. Three formulations have been explained in the paper: functional minimisation, divergence expressions and oriented laplacians. (And a unified expression) which were studied by us. The derived anisotropic diffusion PDE has applications in denoising, inpainting, magnification and flow visualization of colour images. Some of these applications are implemented by us.

2 Theory

The problem of image regularisation can be solved by three models, namely functional minimisation, divergence expression and oriented laplacians.
In these models for vector valued images, a coupling between image channels appears in equations by taking into consideration local vector geometry through structure tensor.

$$\mathbf{G} = \sum_{j=1}^{n} \Delta I_j \Delta I_j^T \quad \text{where } n = 3 \; for \; RGB$$

The structure tensor and it's eigen vectors represent directions of maximum variation. Therefore, the problem can be formulated as minimising the generalized functional

representing the variation in the image using eigen vectors as follows:

$$\min_{I:\Omega \longrightarrow R^n} E(I) = \int_\Omega \psi(\lambda_+, \lambda_-) d\Omega$$

Above equation can be converted to divergence of a diffusion matrix with gradient of the Image channel using Euler-Lagrange theorem, orthogonality of eigen vector and single partial differential equation chain rules as derived in Appendix A of paper [1].

$$\frac{\partial I_i}{\partial t} = \text{div}\left(\mathbf{D}\Delta I_i)\right) \quad (i = 1 \cdots n),$$
$$\mathbf{D} = \frac{\partial \psi}{\partial \lambda_+}\theta_+\theta_+^T + \frac{\partial \psi}{\partial \lambda_-}\theta_-\theta_-^T$$

Above, n is number of channels, θ_+, θ_- ad λ_+, λ_- are the eigen vectors and eigen values respectively of structure tensor.
The oriented Laplacian equation can be written as the following PDE:

$$\frac{\partial I_i}{\partial t} = c_1 I_{i_{\xi\xi}} + c_2 I_{i_{\eta\eta}} = \text{trace}(\mathbf{TH_i})$$

After some derivations a unified expression can be obtained. While implementing this derived expression, some assumptions about images and desired transformations can be made like we want tensor to be isotropic in low variation regions to do smoothing, and it should do anisotropic smoothing along vector edge θ_-, in order to preserve them. Thus, in denoising, inpainting and image magnification, following set of formulae are used:

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

$$\mathbf{T} = f_-\left(\sqrt{\lambda_+^* + \lambda_-^*}\right)\theta_-^*{\theta_-^*}^T + f_+\left(\sqrt{\lambda_+^* + \lambda_-^*}\right)\theta_+^*{\theta_+^*}^T$$

Above eigen parameters are from $\mathbf{G}_\sigma = \mathbf{G} * G_\sigma$, a Gaussian smoothed version of structure tensor $\mathbf{G}$. Finally $I_{i_{(t)}} = I_{i_{(t=0)}} * G^{(T,t)}$.

In **flow visualisation**, the direction of smoothing is entirely independent of the local geometry of the image. It is specified by a separate vector field. So, the update step only requires the Hessian matrix at each pixel from the image. The structure tensor $\mathbf{T}$ need not be calculated in each iteration, it is obtained from the vector field at each pixel $\mathscr{F}$. The image update is given by the formula:

$$\frac{\partial I_i}{\partial t} = trace(\mathbf{TH_i}) \quad (i = 1..n)$$

$$\mathbf{T} = \frac{1}{||\mathscr{F}||}\mathscr{F}\mathscr{F}^T$$

3 Results

3.1 Image Inpainting

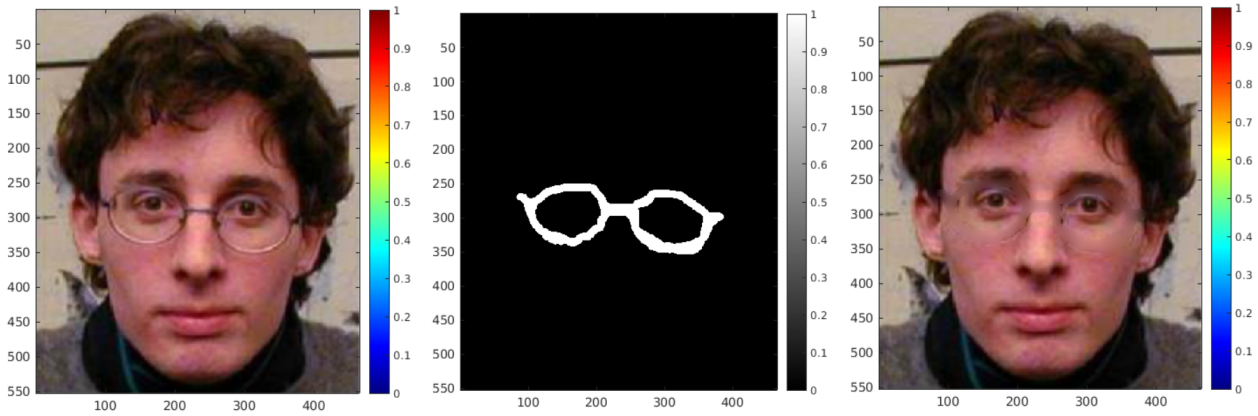


Figure 1: Inpainting with $\sigma = 1$, 10 iteration, window: 21x21 and time=10.0

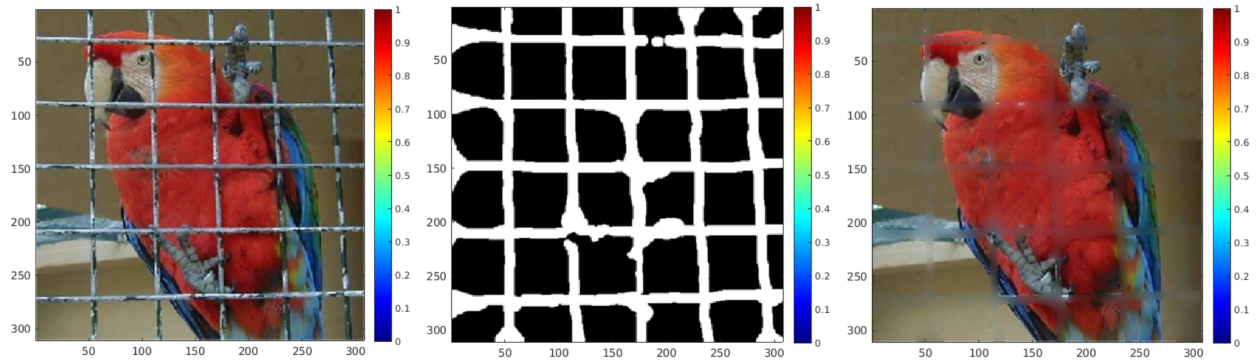


Figure 2: Inpainting with $\sigma = 1$, 10 iteration, window: 21x21 and time=10.0

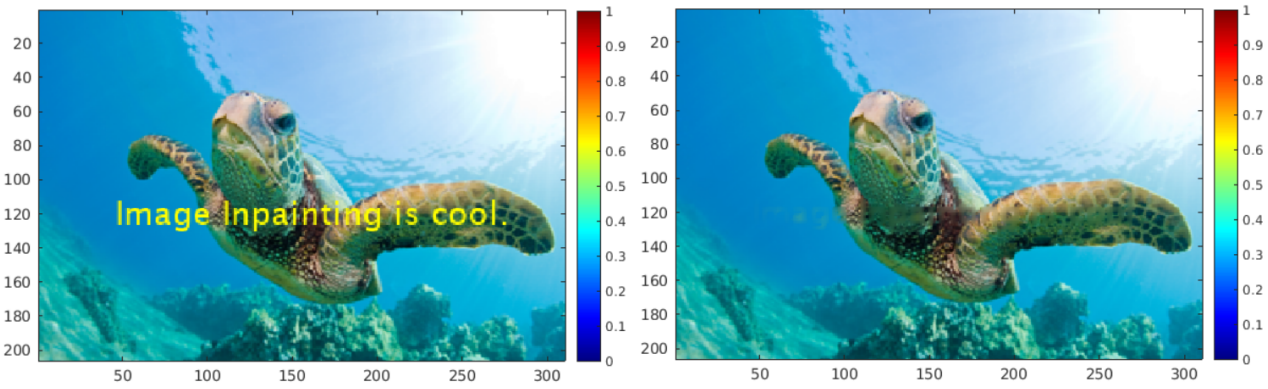


Figure 3: Inpainting with $\sigma = 1$, 10 iteration, window: 21x21 and time=6.0

3.2 Flow Visualisation

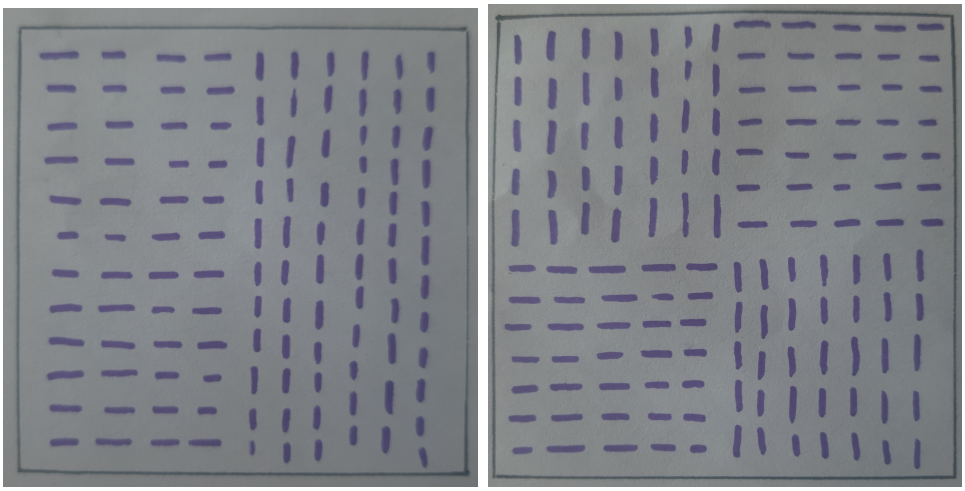
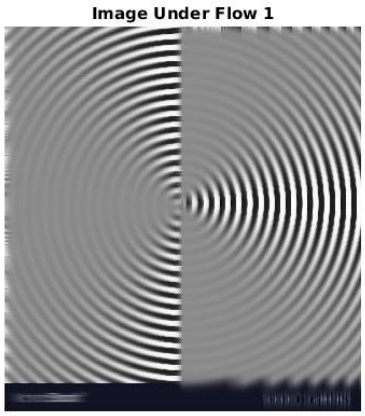


Figure 4: Flow visualisations 1 and 2, field depiction

Figure 5: Flow visualisations (Flow1, Flow2) with of iterations = 100, learning rate = 0.01

Figure 6: Flow visualisations (Flow1, Flow2) with of iterations = 75, learning rate = 0.01



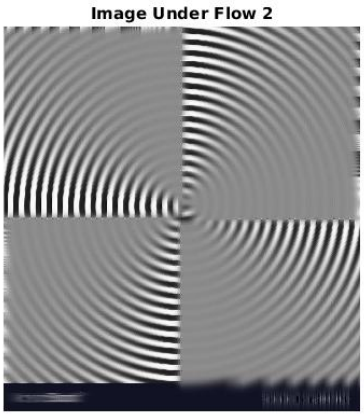
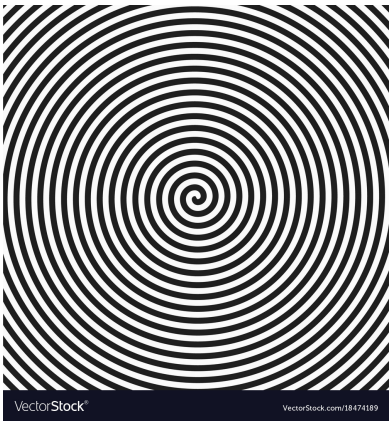


Figure 8: Flow visualisations (Flow1, Flow2) with of iterations = 50, learning rate = 0.01



3.3 Image Restoration

Figure 9: Restoration with 3*3 window with structural similarity 0.982 with the original image

Figure 10: Restoration with 2*2 window with structural similarity 0.982 with the original image

3.4 Denoising

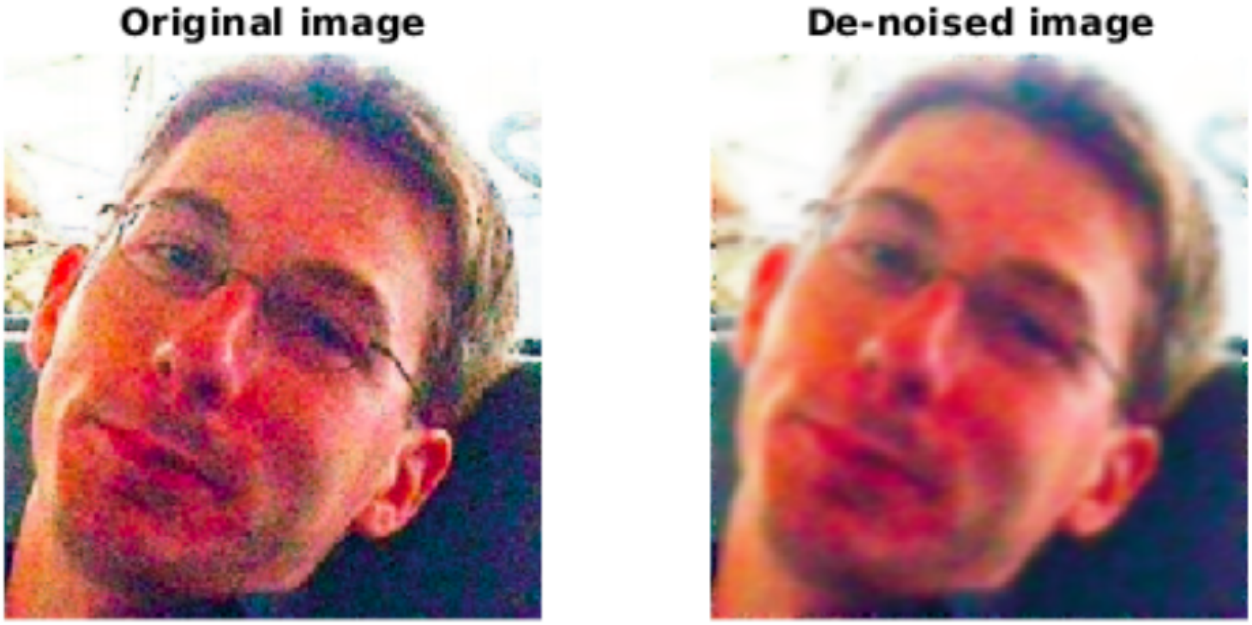


Figure 11: Image denoising with parameters - window size: 7x7, t=5

Figure 12: Image denoising with parameters - window size: 3x3, t=15

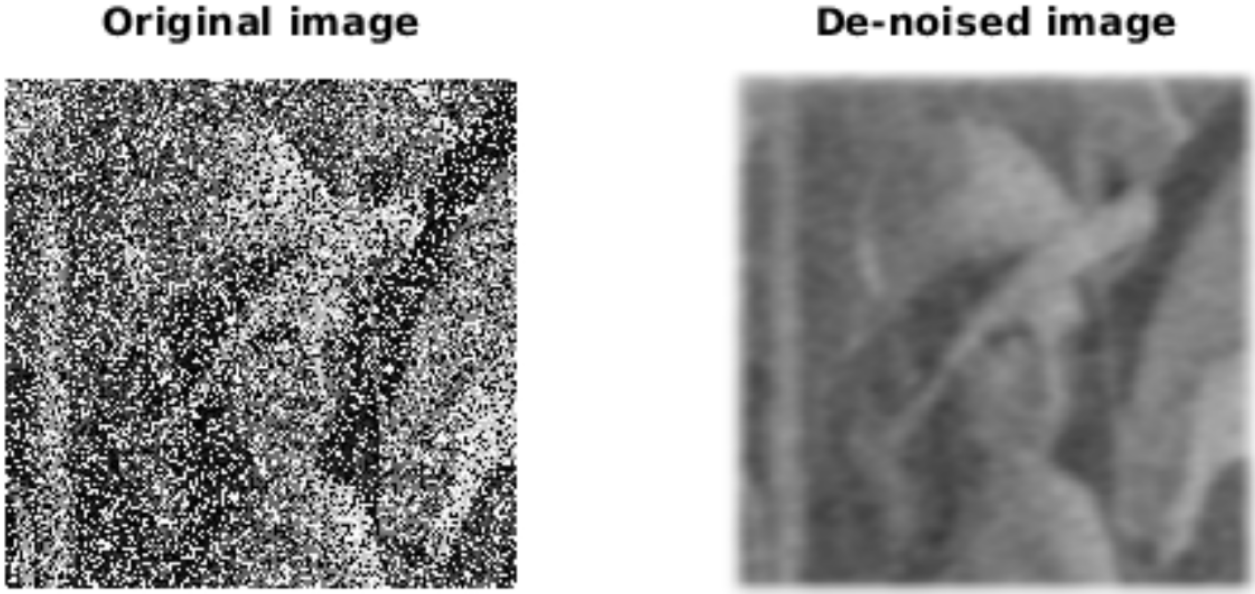


Figure 13: Image denoising with parameters - window size: 11x11, t=5, num_iter=2

4 Observations and Conclusions

- Image Inpainting takes long time to run for larger images and when larger area is under the mask. Fig 10 took 582s, Fig 2 took 1584s and Fig 3 took 93s. Tuning of parameters is difficult since they are many and not all cause significant improvement. Window size must be big when bigger objects are to be removed but then diffusion reduces texture details. As time or number of iterations are increased, masked portion tends to become constant intensity patch.

- Flow visualisation runs quickly, the structure tensor is found once per vector field. A higher number of iterations leads to more regularisation, and a closer resemblance of the image to the vector field.
- IMage denoising runs quickly
- Image Restoration directly uses the impainting function so it also takes quite long but due to downsampling, not as long as impainting

References

[1] D. Tschumperle and R. Deriche, "Vector-valued image regularization with PDE's: A common framework for different applications," IEEE Trans. PAMI, vol. 27, no. 4, pp. 506-517, 2005.