CS663: Fundamentals of Digital Image Processing Project Report

Yash Salunkhe 210020156
Scaria Kochidanadu 210020122
Rishabh Shetty 210040146

Vector-Valued Image Regularization using PDEs

Introduction

Regularization of vector-valued images refers to a set of techniques used to enhance or refine images that are represented as vectors of multiple channels or components. In the context of image processing, vector-valued images are typically color images, where each pixel is represented by multiple values corresponding to different color channels, such as Red, Green, and Blue (RGB) in the case of digital color images.
The main goal of regularization in this context is to reduce noise, smooth out artifacts, or enhance certain features in the image while preserving its structural information and overall quality. Common regularization techniques include denoising (removing noise), image reconstruction (restoring missing or degraded portions of an image), inpainting and flow visualization (highlighting patterns or motion in the image). Regularization methods for vector-valued images often involve mathematical operations, optimization, and the use of variational methods and partial differential equations (PDEs) to achieve the desired image enhancements.

Mathematical formulation

Image regularisation problem can be solved using these 3 interconnected models of functional minimisation, divergence expressions and oriented laplacians.

Functional Minimisation

We employ structure tensor whose eigenvectors represent directions of maximum variation. Therefore, the problem can be formulated as minimising the generalised functional representing the variation in the image using the eigenvectors as follows

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

Divergence expressions

The above equation can be converted to the divergence of a diffusion matrix with gradient of the Image channel by using Euler-Lagrange theorem, heat equations, orthogonality of eigenvectors and simple partial differential equation chain rules as done in the appendix 1 of the original paper.

$$\frac{\partial I_i}{\partial t} = div(D\nabla I) \qquad (i = 1..n)$$

$$\mathbf{D} = \frac{\partial \psi}{\partial \lambda_+}\theta_+\theta_+^T + \frac{\partial \psi}{\partial \lambda_-}\theta_-\theta_-^T$$

where n is the number of channels and θ_+, θ_- are the eigen vectors of the structure tensor. If a ψ exists, eigenvalues of the divergence tensor D can be seen it's gradient and obtaining I_i is easy for every channel.

Oriented Laplacians

The oriented laplacian equation can be written as the following PDE

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

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

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

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

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

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

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

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.

For our experiments we choose, $f_+(s) = \frac{1}{1+s^2}$. and $f_-(s) = \frac{1}{\sqrt{1+s^2}}$

Finally, a high level overview of the algorithm is to convolve the image with a gaussian that solves the PDE of Oriented Laplacians with $\mathbf{T}$ and f as in the above section.

Results

Denoising

As mentioned in the introduction and mathematical formulation, an oriented gaussian kernel is formed using anisotropic diffusion equation PDE which blurs out the features along the eigenvector corresponding to the smaller value not affecting the perpendicular gradient to it

Fig. 1: Denoising with 3x3 neighbourhood, t=5 and 7 iterations

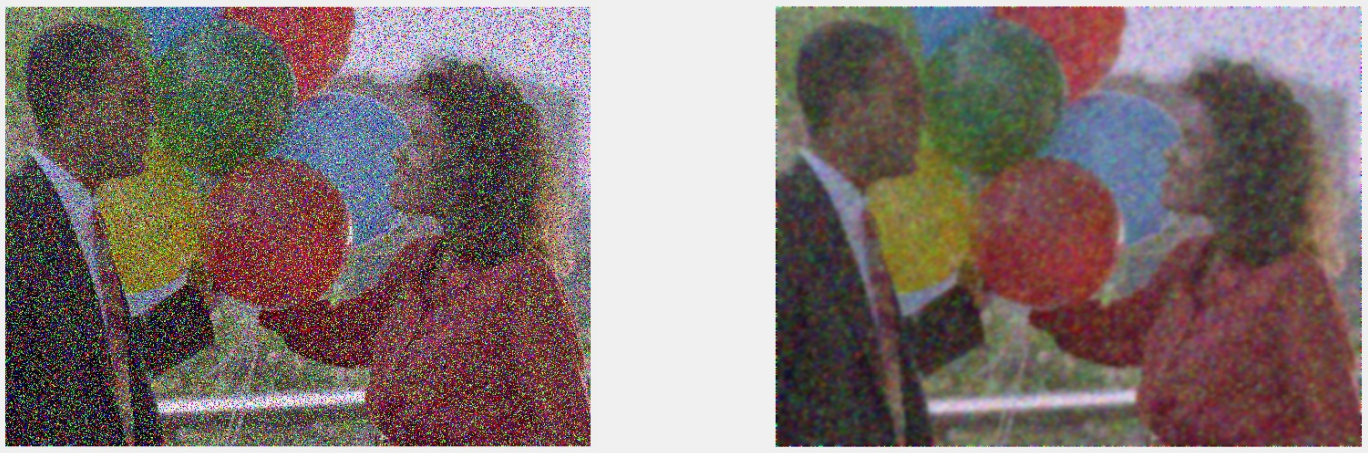

Fig. 2: Denoising with 3x3 neighbourhood, t=5 and 7 iterations

Inpainting

A mask is used to perform blurring only in specified regions which we want to remove, blurring is performed using the oriented gaussian used previously for denoising

Fig. 3: Inpainting with 10x10 neighbourhood and 20 iterations

Fig. 4: Inpainting with 10x10 neighbourhood and 20 iterations

Fig. 5: Inpainting with 10x10 neighbourhood and 20 iterations

Flow Visualization

Following the idea presented in section 7.5 of the paper, we can choose the diffusion relation to follow a specified flow pattern which is used in this section

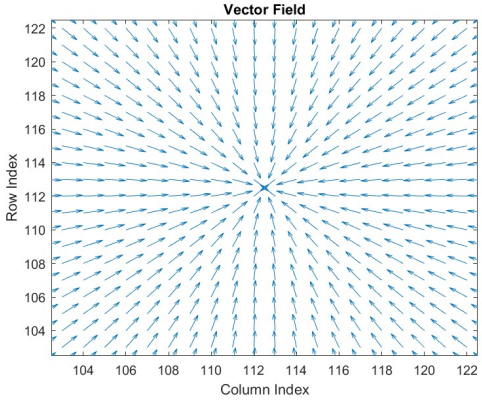

Fig. 6: Flow Field 1

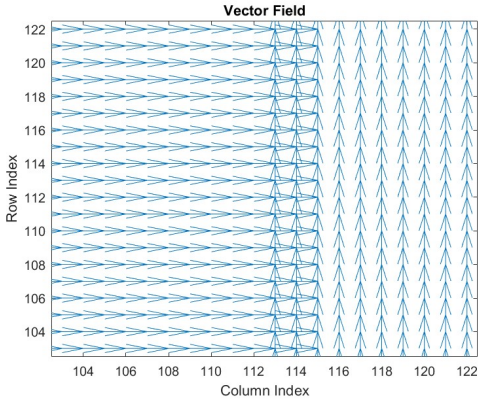

Fig. 7: Flow Field 2

Fig. 8: Flow Field 1

Fig. 9: Flow Field 2

Fig. 10: Flow Field 1

Fig. 11: Flow Field 2

Image Restoration

Fig. 12: Restored image with 50 percent missing pixels

Fig. 13: Restored image with 50 percent missing pixels

Observations and Conclusion

1) The technique proposed performs as good as bilateral filtering for denoising and is computationally less expensive as it uses numerical methods instead of gradient based minimization
2) Inpainting fails when the mask is continuous. By the approach we followed, we need to have a solid neighbourhood around the point and thereby it works well on small radius patches in the image. On increasing the width, the lost information is harder to recover due to the high variability in diffusion.
3) Unlike denoising and inpainting, flow visualization with PDEs doesn't require calculation of Tensor field at each point so with vector manipulations in MATLAB we could very efficiently compute it if the Field vector is not dependent on local variations at each pixel.
4) The three parameters, neighbourhood, time and number of iterations affect the outputs obtained, as we increase the kernel size we observe that more smoothing takes place and sharp features are lost which holds true as per the theory studied in lectures on structure tensor eigenvector properties, beyond a certain number of iterations the image starts to converge, time doesn't affect much in a broader sense but as it tends to infinity the filter acts as a standard gaussian filter performing isotropic smoothing and no regularization takes place if t is kept very small

References

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.