LIC-BASED REGULARIZATION OF MULTI-VALUED IMAGES

David Tschumperlé

Image Team / GREYC (UMR CNRS 6072)
6 Bd du Maréchal Juin, 14050 Caen Cedex. France

ABSTRACT

In this paper, a general multi-valued image regularization method based on LIC's (Line Integral Convolutions [4]) is proposed. From the investigation of recent approaches based on multi-valued diffusion PDE's, we show how a regularization process is naturally decomposed, first as the estimation of its underlying smoothing geometry, and then, as the application of a locally and spatially oriented smoothing. Performing this last part using LIC's significatively improves the overall regularization process both in visual quality and processing time. We illustrate three different applications of our general regularization framework : Color image denoising, inpainting and magnification.

1. INTRODUCTION

Obtaining regularized versions of noisy or scratched data has always been a desirable goal in the fields of computer vision and image processing. It is useful, either to restore images corrupted by noise (which is the most direct application of image regularization) or - more indirectly - as a pre-processing step that eases further analysis of the considered data. Since the pioneering work of Perona-Malik [8], anisotropic diffusion PDE's (Partial Differential Equations) raised a strong interest for this purpose : such equations have the ability to smooth data in a *nonlinear* way, allowing the preservation of important image features (contours, corners or other discontinuities). Thus, many diffusion PDE's have been proposed so far for the restoration of scalar and multi-valued images ([1, 6, 9, 10, 11, 14, 17] and references therein). Despite this wide range of existing formalisms, all methods have something in common : they locally smooth the image along one or several directions that are different at each image point. Typically, the principal smoothing direction is often parallel to the contours within the image, resulting then in an *anisotropic* regularization process.

Thus, defining a correct smoothing behavior is one of the key point of a good regularization algorithm. Recently, authors of [14, 17] proposed general frameworks able to design a specific regularization process from a given underlying local smoothing geometry : one first retrieves the geometry of the structures inside the image (generally by the computation of the so-called *structure tensor field*). Then, a local geometry of the desired smoothing is defined by the mean of a second field of diffusion tensors (depending on the first one). Finally, one step of the smoothing process itself is performed through a specific diffusion PDE.

In this paper, we first review these efficient regularization methods. Then, keeping in mind the idea of separating the regularization process from the design of its smoothing geometry, we propose a new LIC-based framework that regularizes a multi-valued image according to a defined local smoothing geometry. Our method has two main advantages compared to classical PDE implementations : In one hand, it better preserves the orientations of small image structures. In the other hand, it runs up to three times faster.

Finally, we illustrate the effectiveness of our generic LIC-based regularization method, with results on color image restoration, inpainting and non-linear magnification, among all possible applications.

2. PDE-BASED REGULARIZATION

Let us consider a corrupted multi-valued image $\mathbf{I} : \Omega \rightarrow \mathbb{R}^n$ ($n = 3$ for color images) defined on a domain $\Omega \subset \mathbb{R}^2$. We denote by $I_i : \Omega \rightarrow \mathbb{R}$, the particular channel i of the image $\mathbf{I} : \forall \mathbf{X} \in \Omega, \quad \mathbf{I}_{(\mathbf{X})} = \left(I_{1(\mathbf{X})} \;\; I_{2(\mathbf{X})} \;\; \ldots \;\; I_{n(\mathbf{X})} \right)^T$.

Regularization algorithms in [14, 17] consist in computing firstly the smoothed *structure tensor* field $\mathbf{G}_\sigma = \mathbf{G} * G_\sigma$, where G_σ is a 2D gaussian kernel (with a variance σ) and $\mathbf{G} : \Omega \rightarrow \mathrm{P}(2)$ is the field of the symmetric and positive matrices defined as : $\forall \mathbf{X}, \; \mathbf{G}_{(\mathbf{X})} = \sum_{i=1}^{n} \nabla I_{i(\mathbf{X})} \nabla I_{i(\mathbf{X})}^T$.

As noticed in [5, 17], $\mathbf{G}_{\sigma(\mathbf{X})}$ is a good estimator of the local multi-valued geometry of $\mathbf{I}$ at $\mathbf{X}$: its spectral elements give at the same time the vector (color) variations (eigenvalues λ_1, λ_2 of $\mathbf{G}_\sigma$) and the orientation (edges) of the local structures (eigenvectors $\mathbf{u} \perp \mathbf{v}$ of $\mathbf{G}_\sigma$) of $\mathbf{I}$ at each point $\mathbf{X} \in \Omega$. The convolution of $\mathbf{G}$ by G_σ allows the estimation of a more coherent multi-valued geometry.

Starting from this simple geometric measure $\mathbf{G}_\sigma$ of the image, authors of [14, 17] design then a tensor field $\mathbf{T} : \Omega \rightarrow \mathrm{P}(2)$ which defines the desired local smoothing behavior of the regularization process :

$$\forall \mathbf{X}, \quad \mathbf{T}_{(\mathbf{X})} = f_1(\lambda_1, \lambda_2) \, \mathbf{u}\mathbf{u}^T + f_2(\lambda_1, \lambda_2) \, \mathbf{v}\mathbf{v}^T \qquad (1)$$

where $\mathbf{v}$ corresponds to the principal eigenvector of $\mathbf{T}_{(\mathbf{X})}$.

Basically, f_1 and f_2 define the strengths of the desired smoothing along corresponding directions $\mathbf{u}$ and $\mathbf{v}$, at $\mathbf{X}$. The regularization process itself is then done by evolving a diffusion PDE, either

$$\frac{\partial I_i}{\partial t} = \text{div}(\mathbf{T}\nabla I_i) \quad [17] \quad \text{or} \quad \frac{\partial I_i}{\partial t} = \text{trace}(\mathbf{T}\mathbf{H}_i) \quad [14]$$

where $\mathbf{H}_i$ stands for the hessian matrix of $I_{i(\mathbf{X})}$. Comparisons between these two equations is discussed in [14] and is out of the scope of this paper. Nevertheless, we remind the key idea that the smoothing is performed after the design of a *local smoothing geometry defined by a field* $\mathbf{T}$ *of diffusion tensors*. Note also that $\mathbf{T}$ might be updated during the process (after one or several PDE iterations).
Applying such diffusion equations is very time consuming : $\mathbf{I}$ is regularized little by little since PDE-based processing is very local, even with recent and fast implementations [16].

3. LIC-BASED REGULARIZATION

In order to speed up the smoothing process itself, we rather propose to replace the application of diffusion PDE's by a specific LIC-based method. Nevertheless, the first step (i.e. the computation (1) of the tensor field $\mathbf{T}$ defining the local smoothing geometry) is preserved.

3.1. LIC and single direction smoothing

LIC's (Line Integral Convolutions) have been first introduced in [4] as a technique to create a textured representation of a vector field $\mathbf{w} : \Omega \rightarrow \mathbb{R}^2$. The general idea, originally expressed in a discrete form, was to smooth an image $\mathbf{I}^{noise}$ - containing only noise - by averaging the pixel values along the integral lines of $\mathbf{w}$ for all points $\mathbf{X} \in \Omega$. A continuous formulation of a LIC is then :

$$\mathbf{I}^{LIC}_{(\mathbf{X})} = \frac{1}{N}\int_{-\sigma}^{\sigma} f(a)\,\mathbf{I}^{noise}(\mathcal{C}_{(\mathbf{X},a)})\,da \qquad (2)$$

where $f : \mathbb{R} \rightarrow \mathbb{R}$ is an even function (decreasing on $\mathbb{R}^+$, typically a 1D gaussian) and $\mathcal{C} : \Omega \times \mathbb{R} \rightarrow \Omega$ defines the *integral curve* of $\mathbf{w}$ starting from $\mathbf{X}$ and parameterized by $a \in \mathbb{R}$, such that :

$$\begin{cases} \mathcal{C}_{(\mathbf{X},0)} &= \mathbf{X} \\ \frac{\partial \mathcal{C}}{\partial a}(\mathbf{X},a) &= \mathbf{w}(\mathcal{C}_{(\mathbf{X},a)}) \end{cases}$$

The normalization factor is $N = \int_{-\sigma}^{\sigma} f(a)\ da$. The σ parameter controls the smoothing length (size of the neighborhood used on the integral line by the LIC filter). Intuitively, eq.(2) actually smoothes the image along the vector field $\mathbf{w}$. This vector flow visualization problem has been also recently tackled by PDE-based methods in [2, 14]. It is basically based on the regularization methods described in section 2, with a field $\mathbf{T}$ of diffusion tensors defined as : $\forall \mathbf{X} \in \Omega, \quad \mathbf{T}_{(\mathbf{X})} = \mathbf{w}_{(\mathbf{X})}\mathbf{w}^T_{(\mathbf{X})}$.

Note that these smoothing methods are not *strictly* equivalent, but they are based on the same rough idea : smoothing an image along given directions defined by a vector field $\mathbf{w}$. We investigate the theoretical link between diffusion PDE's and LIC-based methods in [15].

3.2. Multi-directional smoothing

The similarity between these two techniques performing single direction smoothing naturally suggests to replace the smoothing phase of general regularization PDE algorithms by a LIC-filtering process. Naturally, this implies that we have to extend the LIC principle (that considers only one *single* direction) to deal with general tensor fields $\mathbf{T} : \Omega \rightarrow \mathrm{P}(2)$ instead of vector fields $\mathbf{w}$. We propose to do so by *averaging multiples LIC's* starting from the same point $\mathbf{X}$, but following different paths.
Let us denote by $\mathbf{U}_\theta$ the vector $\mathbf{U}_\theta = (\cos\theta \quad \sin\theta)^T$. Then, the vector $\mathbf{w}^\theta_{(\mathbf{X})} = \mathbf{T}_{(\mathbf{X})}\ \mathbf{U}_\theta$ verifies :

- If $\mathbf{T}_{(\mathbf{X})}$ is isotropic then $\mathbf{w}^\theta_{(\mathbf{X})} = \alpha\mathbf{U}_\theta$.
- If $\mathbf{T}_{(\mathbf{X})}$ is anisotropic and directed along $\mathbf{U}_\theta$, then $\mathbf{w}^\theta_{(\mathbf{X})} \simeq \alpha\mathbf{U}_\theta$.
- If $\mathbf{T}_{(\mathbf{X})}$ is anisotropic and orthogonal to $\mathbf{U}_\theta$, then $\mathbf{w}^\theta_{(\mathbf{X})} \simeq \vec{0}$.

That can be understood as follows : the more $\mathbf{U}_\theta$ represents a part of $\mathbf{T}$, the higher will be the norm $\|\mathbf{w}^\theta_{(\mathbf{X})}\|$.
For a given θ, we propose to compute a LIC-filtering starting from $\mathbf{X}$ and directed by the vector field $\mathbf{w}^\theta$, then we average all these "atomic" LIC filters for all $\theta \in [0, \pi]$. This range is sufficient to reach the entire plane since LIC's are computed backward and forward along the integral lines $\mathcal{C}^\theta$.
Finally, our regularization equation, computing an anisotropically smoothed version $\mathbf{I}^{regul}$ of an image $\mathbf{I}^{noisy}$ is : $\forall \mathbf{X}$,

$$\mathbf{I}^{regul}_{(\mathbf{X})} = \frac{1}{N}\int_0^\pi \int_{-\sigma\|\mathbf{w}^\theta_{(\mathbf{X})}\|}^{\sigma\|\mathbf{w}^\theta_{(\mathbf{X})}\|} f(a)\ \mathbf{I}^{noisy}(\mathcal{C}^\theta_{(\mathbf{X},a)})\ da\ d\theta \qquad (3)$$

where the normalization factor is $N = \int\int f(a)\ da\ d\theta$,

$$\mathbf{w}^\theta_{(\mathbf{X})} = \mathbf{T}_{(\mathbf{X})}\ \mathbf{U}_{(\theta)} \quad \text{and} \quad f(a) = \exp\left(-\frac{a^2}{2\sigma}\right)$$

σ is a user-defined parameter setting the global strength of the smoothing filter. $\mathcal{C}^\theta : \Omega \times \mathbb{R} \rightarrow \Omega$ represents the integral line of the field $\mathbf{w}^\theta$ starting from $\mathbf{X}$ and parameterized by a :

$$\begin{cases} \mathcal{C}^\theta_{(\mathbf{X},0)} &= \mathbf{X} \\ \frac{\partial \mathcal{C}^\theta}{\partial a}(\mathbf{X},a) &= \mathbf{w}^\theta(\mathcal{C}^\theta_{(\mathbf{X},a)}) = \mathbf{T}(\mathcal{C}^\theta_{(\mathbf{X},a)})\ \mathbf{U}_{(\theta)} \end{cases}$$

It is easy to verify that :

- The equation (3) respects the extremum principle, since only pixel averaging is performed.
- For points $\mathbf{X} \in \Omega$ where $\mathbf{T}_{(\mathbf{X})}$ is isotropic, the smoothing is also isotropic (averaging along all directions of the

plane), since $\forall\theta$, $\|\mathbf{w}^{\theta}_{(\mathbf{X})}\|$ is constant.
• For points $\mathbf{X}$ where $\mathbf{T}_{(\mathbf{X})}$ is highly anisotropic, the smoothing is done only along the principal eigenvector of $\mathbf{T}_{(\mathbf{X})}$.
These properties are also verified by most of the proposed anisotropic regularization PDE's, and ensure that the smoothing process is indeed driven by the tensor field $\mathbf{T}$.
• The algorithme is very stable, even with very large time step dt, since only pixel averaging is performed. Moreover, the rotational invariance of the scheme is ensured from a numerical point of view.

4. APPLICATIONS

The Fig.1 illustrates different applications of our LIC-based regularization algorithm on color images. Processing time is mentionned on the figure captions. Experiments have been done on a Sun Sparc 1.5 Ghz. Results are obtained quite fastly compared to PDE techniques. Actually, one often needs only one iteration to achieve the regularization since the LIC-based scheme is stable by construction. For our experiments, we defined $\mathbf{T}$ with eq.(1) and

$$f_{1/2}(\lambda_1, \lambda_2) = \frac{1}{(\epsilon+\lambda_1+\lambda_2)^{p}_{1/2}} \tag{4}$$

where $\epsilon = 0.01$, $p_1 = 0.5$ and $p_2 = 2$ (for denoising purpose) or $p_2 = 10$ (for inpainting and magnification purposes). The discretization is done with classical finite differences to compute the smoothed structure tensor field $\mathbf{G}_\sigma$ (updated at each iteration), and discrete sums to approximate the two integrals of (3). The discretization of θ is quite large ($d\theta = 30^o$) while a is more precisely approximated ($da = 0.2$). We applied eq.(3) for :
• **Color image denoising** (Fig.1a,b,c) : Only one iteration of our algorithm has been necessary to obtain the results. Note how the noise is removed while the variously oriented image structures are well preserved (rotational invariance property of the scheme).
• **Color image inpainting** (Fig.1d) : For this application, we applied eq.(3) only on the points inside a user-defined mask (a checkerboard mask in our case). 100 iterations have been needed to get the results.
• **Non-linear interpolation for magnification** (Fig.1e) : These two examples shows how our LIC-based regularization technique (3) can perform super-resolution. A linear interpolation of the small image is regularized while keeping the know points unchanged (similar to inpainting with a very sparse grid mask).
You can get more results and the algorithm executable at :
`http://www.greyc.ensicaen.fr/~dtschump/greycstoration/`

5. CONCLUSION

We have proposed a very generic and efficient regularization algorithm acting on multi-valued images. It gathers the best of classical filtering techniques used in the field of diffusion PDE's and LIC's. This allows us to propose a new regularization framework that improve result quality and processing time. Our LIC-based method is simple to implement and potentially handles a very wide range of image processing applications.

6. REFERENCES

[1] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: PDE's and the Calculus of Variations*, vol. 147 of *Applied Mathematical Sciences*.Springer-Verlag, January 2002.

[2] J. Becker, T. Preusser, and M. Rumpf. *PDE methods in flow simulation post processing.* Computing and Visualization in Science, 3(3):159–167, 2000.

[3] M.J. Black, G. Sapiro, D.H. Marimont, and D. Heeger. *Robust anisotropic diffusion.* IEEE Transaction on Image Processing, 7(3):421–432, 1998.

[4] B. Cabral. *Imaging vector fields using line integral convolution.* Computer Graphics Proceedings, p.263–270, 1993.

[5] S. Di Zenzo *A note on the gradient of a multi-image.* Computer Vision, Graphics and Image Processing, 33:116-125, 1986.

[6] R. Kimmel, R. Malladi, and N. Sochen. *Images as embedded maps and minimal surfaces: movies, color, texture, and volumetric medical images.* International Journal of Computer Vision, 39(2):111–129, September 2000.

[7] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994.

[8] P. Perona and J. Malik. *Scale-space and edge detection using anisotropic diffusion.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629–639, July 1990.

[9] L. Rudin, S. Osher, and E. Fatemi. *Nonlinear total variation based noise removal algorithms.* Physica D, 60:259–268, 1992.

[10] G. Sapiro. *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, 2001.

[11] G. Sapiro and D.L. Ringach. *Anisotropic diffusion of multi-valued images with applications to color filtering.* IEEE Transactions on Image Processing, 5(11):1582–1585, 1996.

[12] N. Sochen, R. Kimmel, and A.M. Bruckstein. *Diffusions and confusions in signal and image processing.* Journal of Mathematical Imaging and Vision, 14(3):195–209, 2001.

[13] C. Tomasi and R. Manduchi. *Bilateral filtering for gray and color images.* Proceedings of the IEEE International Conference on Computer Vision, pages 839–846, January 1998.

[14] D. Tschumperlé, R. Deriche *Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications.* Computer Vision and Pattern Recognition, Madison/USA, June 2003.

[15] D. Tschumperlé *Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's.* Research Report "Les Cahiers du GREYC", No 05/01. Equipe IMAGE/GREYC (CNRS UMR 6072), Février 2005.

[16] J. Weickert, K.J. Zuiderveld, B.M ter Haar Romeny, W.J. Niessen. *Parallel Implementations of AOS Schemes : A Fast Way of Nonlinear Diffusion Filtering*. ICIP'97 (3:396-399), Santa Barbara, Oct 1997.

[17] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, Stuttgart, 1998.

(a) Removing grain noise from a 512×512 color image.
(detail, 1 iteration, processing time : 19 seconds)

(b) Removing pattern noise from a 555×367 color image.
(detail, 1 iteration, processing time: 11 seconds)

(c) Removing JPG artefacts from a 300×300 color image.
(1 iteration, processing time : 13 seconds)

(d) 290×290 color image inpainting.
(100 iterations, processing time : 1 minute 26 s)

(e) Non-linear image magnification (1 iteration, processing time : respectively 20 seconds)
From left to right : original color image, nearest-neighbor interpolation, bicubic interpolation, our LIC-based interpolation (3).

Fig. 1. Results of our LIC-filtering method applied to various image processing issues.