Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 53(67), Fascicola 2, 2008

An Improved Method for Directional Image Smoothing Based on Structure Tensors and Vector Field Visualisation Techniques

Cosmin Ludusan[1,2], Olivier Lavialle[1], Sorin Pop[1,2], Romulus Terebes[2], Monica Borda[2]

Abstract – We propose an alternative diffusion technique, starting from a tensor-based method that can perform both isotropic and anisotropic smoothing using as directional information the eigenvectors of the image structures. The method employs two tensors, a structure and a diffusion tensor respectively, the novelty consisting in the manner in which we design the diffusion tensor. The developed method allows a fast implementation and a reduced number of time iterations, being discretized by a trace-based method. Testing is performed both on synthetic and real images, with quality measurements and remarks.

Keywords: diffusion, PDEs, trace operation, tensor fields, vector field visualisation.

I. INTRODUCTION

The first topic discussed in this paper is related to the notion of isotropic and anisotropic diffusion based on time-evolving PDEs. The starting point of our method is described in [1] where a multi-channel image is considered to be a RGB colour image, where each colour channel is viewed as a separate image. The diffusion technique can be implemented using the traditional numerical schemes described in [13] or by an alternative method proposed in [15], where the evolving PDE can be viewed as the convolution between the input image and a time depending, diffusion tensor-oriented Gaussian kernel, see also [1, 2, 3]. The experimental part was implemented using a derived form of the classical PDE equation, based on the trace operator, its equivalence being demonstrated in [15, 17]. A more in-depth discussion about this will follow in the next paragraph and in the "Results and Remarks" paragraph results of the numerical implementation of this method will be presented as such.

In order to have a robust diffusion algorithm, insensitive to edges and corners as the one described in [1, 2, 3] we also need to develop a way of steering the diffusion process according to the structure vectors. An interesting method of mapping a vector field was first described in [4] with the purpose of vector field visualization in order to overcome the resolution-dependent problems inherent to the previous methods. The idea is similar to the one presented in [1, 2, 3], that is, to develop an adapted model of vector mapping, similar to the one described in [4] and to use this vector map to steer the diffusion process along vector lines, also known as LICs (Line Integral Convolutions). This part of the diffusion process will be discussed in further detail in paragraph 3.

The experimental results of our work so far, along with remarks regarding these results, are presented in paragraph 4. These results include the practical implementation of the theoretical aspects presented in paragraphs 2 and 3. The tests presented in paragraph 4 were performed only on 8-bit grey level images, extensions to RGB images and 3D space will be performed as a further developing step.

The final paragraph of this article offers an insight of our future work, some of the ideas that we intend to investigate in the future along with emerging concepts regarding both diffusion and fusion processes. The ultimate goal is to develop a robust theoretical model for a diffusion process insensitive to image structures and then adapt this model into a fusion one, thus obtaining a hybrid process that performs fusion with diffusion-like characteristics, mainly denoising performed in parallel to the main fusion process.

[1] Equipe Signal et Image, UMR LAPS 5131, Av. Du Dr. Schweitzer BP 99, 33402 Talence, France, {olivier.lavialle, sorin.pop}@laps.ims-bordeaux.fr

[2] Technical University of Cluj-Napoca, 26-28 Baritiu Street, 400027, Cluj-Napoca, Romania, {cosmin.ludusan, romulus.terebes, monica.borda}@com.utcluj.ro

II. MULTI-CHANNEL PDE-BASED DIFFUSION

The idea of multi-channel diffusion is related to colour images that can be interpreted as separate channels (i.e. R, G and B) and can be processed separately or together. A multi-channel image can be mathematically viewed as a mapping $I:\Omega \to R^n$ (where in this case n is considered to be the number of channels, thus n=3 for RGB colour images). The domain Ω in this case is considered to be: $\Omega \subset R^2$. A single channel can be defined as $I_i:\Omega \to R$ where the value of each pixel of the multi-channel image is defined as follows [1]: $\forall X=(x,y)\in\Omega,\ I_X=(I_X^1, I_X^2, \ldots, I_X^n)^T$.

The following discussions will focus on the methods used to describe the geometry of a grey-level image. Therefore, from this point onward, for the simplicity of our discussion, we will consider the original image corrupted by noise as being $I:\Omega \to R$ and $\Omega \subset R^2$.

Before performing the diffusion on an image corrupted by noise, we need to make sure that we have a robust method of describing its geometry. A good way of doing that is by computing the local geometry based on the eigenvectors and eigenvalues of the structures of the image. An efficient way of containing that information in a unified manner is to compute a structure tensor G based on the local gradient computed in each pixel of the image I [10]:

$$\forall X\in\Omega,\ G_X=\nabla I_X\cdot\nabla I_X^T, \quad where \quad \nabla I_X=\begin{pmatrix}\frac{\partial I_X}{\partial x}\\ \frac{\partial I_X}{\partial y}\end{pmatrix} \tag{1}$$

The discretization of the partial derivatives used in computing the gradient can be done in a straightforward manner by using central finite differences, described in detail in [13], or alternatively by means of a convolution mask, also detailed in [13]. By designing the structure tensor in this manner, we implicitly contain the information regarding the eigenvectors and eigenvalues also G can be equivalently written as follows:

$$\forall X\in\Omega,\ G_X=\lambda_-\cdot v_-\cdot v_-^T+\lambda_+\cdot v_+\cdot v_+^T \tag{2}$$

Where v_- and v_+ are two eigenvectors (orthogonal vectors), unit vectors of R^2, computed in each pixel X of the image and corresponding to the local minimum and maximum variations of image intensities in that pixel. λ_- and λ_+ are the corresponding eigenvalues (positive values), they measure the effective variations of the image intensities along the corresponding eigenvectors and are related to the local strength of an edge.

In order to obtain a more coherent geometry, a smoothed version of the structure tensor G_X is computed by performing the convolution between G_X and K_σ which is a 2D Gaussian kernel of the form:

$$K_\sigma(a,b)=\frac{1}{2\pi\sigma^2}\exp(-\frac{a^2+b^2}{4\sigma^2}) \tag{3}$$

An alternative way of applying the convolution kernel is to apply a 1D Gaussian kernel of the form: $K_\sigma(a)=\frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{a^2}{2\sigma^2})$ first on the horizontal direction of the structure tensor and then on the vertical direction of the intermediary structure tensor. In this way we obtain a smoothed structure tensor G_X^σ, which better describes the local geometry of the image I at point X.

Based on the local geometry retrieved from the smoothed structure tensor, [1] proposes the design of a field $T:\Omega\to P(2)$ comprised of diffusion tensors, used to steer the direction process. T also depends on the local geometry of the image I and is defined in the following manner:

$$\forall X\in\Omega,\ T_X=f_-(\lambda_-,\lambda_+)\cdot v_-\cdot v_-^T+f_+(\lambda_-,\lambda_+)\cdot v_+\cdot v_+^T \tag{4}$$

In other words $f_{-/+}:R^2\to R$ represent two functions that set the strength of the smoothing along the two directions given by the eigenvectors v_- and v_+. The classical definition of $f_{-/+}$ can be found in [1]:

$$f_-=\frac{1}{(1+\lambda_-+\lambda_+)^{p_1}}\ and\ f_+=\frac{1}{(1+\lambda_-+\lambda_+)^{p_2}} \quad p_1\le p_2 \tag{5}$$

On the other hand we proposes a new approach in defining these two functions, one that better preserves the dependence on the two eigenvalues that basically indicate the strength of the two eigenvectors:

$$f_-=\left(\frac{\lambda_+}{\lambda_-+\lambda_+}\right)^{p_1}\ and\ f_+=\left(\frac{\lambda_-}{\lambda_-+\lambda_+}\right)^{p_2} \quad p_1\le p_2 \tag{6}$$

p_1 and p_2 are two control parameters used to better control the diffusion process. In paragraph 5 we will present some test images that show that by choosing the functions this way we can better preserve the overall geometry of the initial image while still performing a good denoising. Also the solution's convergence is faster than with the classical definition given in [1]. The smoothing behaviour according to the diffusion tensor T can be interpreted as follows: if the pixel X is located on the edge of an image structure, λ_+ is high therefore the smoothing in X will be performed mostly on the v_- direction since $f_+ << f_-$ (in this case we deal with an anisotropic diffusion); the smoothing strength in this case is inversely proportional to the contour strength. The alternative is that X is located in a homogenous region of the image, thus λ_+ is low, and in this case the smoothing is performed in all possible directions, approaching an isotropic diffusion process. In this way a controlled process that can perform both isotropic and anisotropic diffusion is obtained, the type of the diffusion being controlled by its own parameters, unified under T.

The classical diffusion equation can be written as follows:

$$\frac{\partial I}{\partial t} = div(T \cdot \nabla I) \tag{7}$$

An alternative equation was demonstrated and proposed by [1] based on the *trace* operator:

$$\frac{\partial I}{\partial t} = trace(T \cdot H) \text{ with } H = \begin{pmatrix} \frac{\partial^2 I}{\partial x^2} & \frac{\partial^2 I}{\partial x \partial y} \\ \frac{\partial^2 I}{\partial x \partial y} & \frac{\partial^2 I}{\partial y^2} \end{pmatrix} \tag{8}$$

This formulation can be seen as a small convolution applied around each X with a Gaussian mask K_t oriented by the tensor T [1]:

$$K_t^T(X) = \frac{1}{4\pi t}\exp\left(-\frac{X^T T^{-1} X}{4t}\right), \quad where\ X = (x, y) \tag{9}$$

The simplest diffusion process is the linear and isotropic diffusion that is equivalent to a convolution with a Gaussian kernel. The similarity between such a convolution and the heat equation was proved by Koenderink [18]:

$$\frac{\partial I}{\partial t} = div(c(x, y, t)\nabla I) \tag{10}$$

The *pros* and *cons* of this formulation are given in [1]. The results presented in paragraph 4 are obtained using this discretization along with the proposed T. The practical implementation of the method can be done either by computing only once the structure tensor and the diffusion tensor respectively (based on the original input image) which reduces the overall computational cost, or by computing the two tensors with each time step (the discretized values of *dt*) on the updated version of the input image, which increases the computational costs, but helps in preserving the image geometry.

III. VECTOR FIELD VISUALISATION

One way of steering a diffusion process is by means of a diffusion tensor obtained from a structure tensor of the original image, assumed corrupted by noise. As we have already discussed in the previous paragraph the diffusion can be effectively steered by appropriately choosing T i.e. its functions $f_{-/+}$. Even if the results are pertinent, they still present an over-smoothing behaviour around edges and corners. The work in [1] proposes as a solution to counteract these unwanted effects the steering of the diffusion process along pre-computed paths inside the vector field. A seminal work in this direction is represented by [4], where a method for visualising vector fields irrespective of the field resolution is presented, named LIC (Line Integral Convolution). The method in its entirety does not represent the subject of the present paper, but the method of computing the sub-pixel coordinates of the advections in each pixel represents a key element in solving our problem of diffusion steering. The reason why this method is important is that it can compute vector field directions at a sub-pixel level, thus being very receptive to small variations in the vector field, variations usually found at structure corners or edges. Having a method that can follow this type of variations accurately is of paramount importance in the development of robust diffusion mechanism, and by using the diffusion approach described by [1, 2, 3] we can envision a unified formulation for a future PDE-based fusion method.

The model proposed in [4] represents a model designed for vector field visualisation, thus we need to define an adapted version of it in order to suit our needs.

In defining a model for constructing the positive and negative advections starting from an image point X like the one represented in Fig. 1 we have to define a method of computing the next step of the advection according to the orientation of the vector in the current cell.

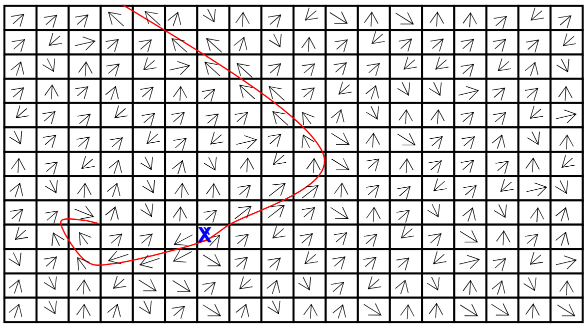


Fig. 1 Visual representation of 2D vector field and a local streamline starting from cell X spreading in the positive and negative direction according to the orientation of each vector from each cell it passes.

A simplified model of doing just that is presented in [4]. In order to use this technique for diffusion steering we need a more robust model:

$$\begin{aligned}
&P_0 = (x + 0.5; y + 0.5) \\
&P_i = P_{i-1} + \frac{V(\lfloor P_{i-1} \rfloor)}{\|V(\lfloor P_{i-1} \rfloor)\|}\Delta s_{i-1} \\
&V(\lfloor P_{i-1} \rfloor) = the\ vector\ with\ the\ lattice\ coordinates \\
&in\ (\lfloor P_{i-1}^x \rfloor, \lfloor P_{i-1}^y \rfloor) \\
&\Delta s_{i-1} = \min(s_{i-1}^{top}, s_{i-1}^{bottom}, s_{i-1}^{left}, s_{i-1}^{right}) \\
&s_{i-1}^{side} = \begin{cases} \infty & if\ V(\lfloor P_{i-1} \rfloor) \| side \\ \infty & if\ \frac{P_{i-1}^e - P_{i-1}^c}{V_{i-1}^c} \le 0 \\ \frac{P_{i-1}^e - P_{i-1}^c}{V_{i-1}^c} & if\ \frac{P_{i-1}^e - P_{i-1}^c}{V_{i-1}^c} > 0 \end{cases} \\
&where\ side \in \{top, bottom, left, right\} \\
&P_{i-1}^e = \lfloor P_{i-1}^c \rfloor + 1, \quad (e;c) \in \{(top; y), (right; x)\} \\
&P_{i-1}^e = \lfloor P_{i-1}^c \rfloor - 1, \quad (e;c) \in \{(bottom; y), (left; x)\}
\end{aligned} \tag{11}$$

Here, the {top, bottom, left, right} represents the system of reference with respect to the vector corresponding to the point X, when the assumption that a pixel is of square shape is made. Equation (11) represents the positive advection starting from X and as it can be seen from the formulation of P_i the coordinates have sub-pixel values since they do not always lay in the discreet lattice coordinate system, hence the sub-pixel accuracy of the method. The negative advection is obtained in a similar way, with the following differences:

$$
\begin{aligned}
P_0' &= P_0 = (x+0.5; y+0.5) \\
P_i' &= P_{i-1}' - \frac{V\left(\left\lfloor P_{i-1}' \right\rfloor\right)}{\left\| V\left(\left\lfloor P_{i-1}' \right\rfloor\right) \right\|} \Delta s_{i-1}'
\end{aligned} \tag{12}
$$

In [4] the complete algorithm for visualizing the vector field based on the streamlines obtained using eq. (11) and (12) is presented, but as we have already pointed out we are interested only in obtaining the streamline coordinates and will be used in a future work to steer diffusion along the image structures described by the eigenvectors (v_- describe the structure edges and contours).

IV. RESULTS AND REMARKS

In this paragraph we will start by presenting the results of our diffusion method based on the *trace* discretization, followed by a comparative test with the base method presented in [1], where the diffusion tensor T is built in a different manner. Another important step that will be tested is the vector coordinates method that will be used in our future work in developing a robust fusion method.

The first set of images, represented in Fig. 2, is the results of the trace-based PDE implementation suggested in [1] as an alternative to the classical diverge-based method where the diffusion tensor T was computed according to equations (4) and (6):

a) b)

Fig. 2 a) Original *Boats* image with Gaussian white noise of RMSE = 25; b) Denoised *Boats* image using T from eq. (4) and (6) the trace-based discretization and 30 iterations with p_1=p_2=1, dt=0.1 and σ=1, RMSE=14.22, PSNR=24.83[dB];

As it can be seen from Fig. 2 the proposed diffusion method performs well in removing noise without creating artefacts in the process and using a reduced number of iterations, thus leading to convergence in a short time interval. Similar results can be obtained using as few as 10 iterations. The anisotropy parameters (p_1 and p_2) were deliberately set for favouring edge preservation and, in this case, some artefacts are created on the background of the image.

Fig. 3 represents a set of images obtained using the definition of T given by equations (4) and (6) compared to the images obtained using the definition of T given in [1]. The original image of *Boats* with Gaussian white noise can be found in Fig. 2a.

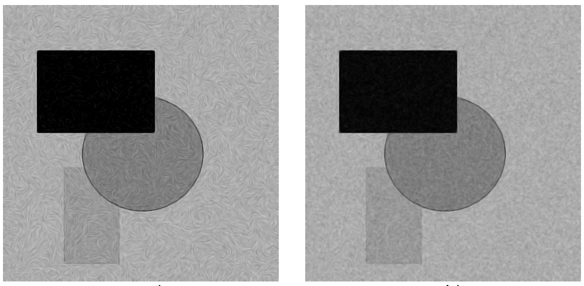
a) b)

Fig. 3 a) *Boats* denoised image using the proposed method for T after 30 iterations with p_1=p_2=3, dt=0.1 and σ=5, RMSE=12.65, PSNR=25.84[dB]; b) *Boats* denoised image using the proposed method for T after 43 iterations with p_1=p_2=3, dt=0.1 and σ=4, RMSE=12.47, PSNR=25.96[dB];

c) d)

Fig.3 c) *Boats* denoised image using the [1] method for T after 30 iterations with p_1=0.1 p_2=1.2, dt=0.1 and σ=4, RMSE=13.51, PSNR=25.27[dB]; b) *Boats* denoised image using the [1] method for T after 80 iterations with p_1=0.1 p_2=1.2, dt=0.1 and σ=4, RMSE=12.72, PSNR=25.79[dB];

As it can be seen from the tests presented in Fig. 3, the proposed method based on equations (4) and (6) needs fewer iterations (i.e. half) to smooth the image to the same degree or even higher as the original method presented in [1]. Another set of tests can be performed on synthetic images (with Gaussian noise added of RMSE=25), like the ones presented in Fig. 4:



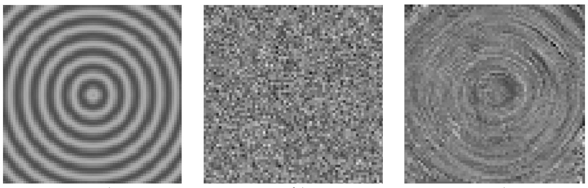
a) b)

Fig.4 a) *Shapes* denoised image using the [1] method for T after 40 iterations with p_1=0.1 p_2=1.2, dt=0.1 and σ=4, RMSE=8.08, PSNR=26.6[dB]; b) *Shapes* denoised image using the proposed method for T after 20 iterations with p_1=p_2=1, dt=0.1 and σ=4, RMSE=7.87, PSNR=26.83[dB];

The number of iterations used in the testing process was chosen in order to provide a similar PSNR and also to keep the 1:2 ratio between the two methods, thus proving the proposed method as being twice as fast. The time evolution is interpreted as being dt * *number of iterations,* therefore for dt=0.1 and *number* *of iterations*=40, time evolution would be 4 seconds, the actual computing time depending on the PC speed usually being higher than the mathematical time evolution.

The stopping time for each method was determined experimentally as in most similar PDE based approaches in the literature.

The vector visualisation problem discussed in paragraph 3 is represented in Fig. 4, where we use the method described in [4] for visualising the 2D vector field with the streamline coordinates obtained using our model described by equation (11).



a) b) c)

Fig. 5 a) Source image for extracting the 2D vector field, in this case the eigenvectors v_- defining the image structures and edges;
b) Gaussian white noise image used for vector representation;
c) Vector field visualisation;

An alternative method of visualising the same vector field (Fig. 5a) is by distorting a second image according to this field, this method was also tested in [4] and the results can be viewed in fig. 6b.

a) b)

Fig. 6 a) *Lena* original image used for the vector field visualisation; b) *Lena* distorted image according to the vector field extracted from the synthetic image from fig. 5a;

The vector visualisation technique is considered to be the first step in our future work which includes developing a new numerical scheme for our diffusion method, steered by vector streamlines, obtained theoretically using equation (11).

V. FUTURE WORK

Our ultimate goal is to develop a robust fusion method derived from a diffusion one, since we are interested in combining the effects of fusion with the ones of diffusion thus obtaining a simpler faster method that can perform both denoising and image restoration at the same time. So far we have defined, implemented and tested diffusion methods along with vector field streamlines generation (intended for use in steering fusion and diffusion process alike). The next step is to develop a mathematical model that unifies the diffusion equations along with the LIC method (vector streamlines method), finally obtaining a fusion model with robust edge preservation and denoising capabilities.

ACKNOWLEDGEMENTS

This work was funded by the Romanian Agency UEFISCSU within the PN 2 IDEI no.908 and 909 01.10.2007 research grants.

REFERENCES

[1] D. Tschumperle. *Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's*, International Journal of Computer Vision, IJCV(68), No 1, June 2006, pp.65-82.

[2] D. Tschumperle. *Curvature-Preserving Regularization of Multi-Valued Images using PDE's*, European Conference on Computer Vision (ECCV'2006).

[3] D. Tschumperle. *LIC-Based Regularization of Multi-Valued Images*, IEEE International Conference on Image Processing (ICIP), September 2005.

[4] B. Cabral and L.C. Leedom. *Imaging vector fields using line integral convolution.* SIGGRAPH'93, in Computer Graphics Vol.27, pp.263–272, 1993.

[5] S. Pop, R. Terebes, M. Borda, O. Lavialle *Low-level fusion: a PDE-based approach*, International Conference on Information Fusion, Fusion2007, 9-12, Québec, Canada, July 2007.

[6] S. Pop, O. Lavialle, M. Donias, R. Terebes, M. Borda S. Guillon, N. Keskes *A PDE-Based approach to 3D seismic data fusion*, accepted by IEEE Transactions on Geoscience and Remote Sensing – Data Fusion Special Issue, Volume 46, Issue 5, May 2008 Page(s):1385 – 1393.

[7] D.A. Socolinsky, L.B. Wolff *Multispectral image visualization through first-order fusion*, IEEE Transaction on Image Processing, vol. 11(8), pp. 923-931, 2002.

[8] S. John, M.A. Vorontsov *Multiframe selective information fusion from robust error estimation theory*, IEEE Transaction on Image Processing, vol. 14(5), pp. 577-584, 2005.

[9] C. Wang, Q. Yang, X. Tang, Z-F. Ye *Salience Preserving Image Fusion with Dynamic Range Compression*, IEEE International Conference on Image Processing, ICIP 2006, pp. 989-992, 2006.

[10] S. Di Zenzo, *Note on the gradient of a multi-image*, Computer Vision, Graphics, And Image Processing, vol. 33, pp. 116-125, 1986.

[11] P. Perona, J. Malik *Scale space and edge detection using anisotropic diffusion*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.12, no.7, pp.629-639, 1990.

[12] J. Weickert *Coherence enhancing diffusion*, International Journal of Computer Vision , no.31, pp. 111-127, 1999.

[13] R. Terebes *Diffusion directionnelle. Applications a la restauration et a l'amelioration d'images de documents anciens*, These de doctorat, Universite Bordeaux I et Technical University of Cluj-Napoca, 2004.

[14] D. Tschumperle and R. Deriche. *Diffusion PDE's on Vector-Valued images.* IEEE Signal Processing Magazine, 19(5):16–25, 2002.

[15] D. Tschumperle and R. Deriche. *Vector-valued image regularization with PDE's : A common framework for different applications*. RR 4657, INRIA Sophia-Antipolis, December 2002.

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

[17] D. Tschumperle *PDE's Based Regularization of Multi-valued Images and Applications.* PhD Thesis, Universite de Nice-Sophia Antipolis/France, December 2002.

[18] J. Koenderink, *The structure of images*, Biological Cybernetics, Vol.50, pp. 363-370, 1984.