FLOW ENHANCING LINE INTEGRAL CONVOLUTION FILTER

Tolga Birdal, Emrah Bala

Befunky Inc.

ABSTRACT

Visualization of vector fields is an operation used in many fields such as science, art and image processing. Lately, line integral convolution (LIC) technique [1], which is based on locally filtering an input image along a curved stream line in a vector field, has become very popular in this area because of its local and robust characteristics. For smoothing and texture generation, used vector field deeply affects the output of LIC method. We propose a new vector field based on flow fields to use with LIC. This new hybrid technique is called flow enhancing line integral convolution filtering (FELIC) and it is highly capable of smoothing an image and generating high fidelity textures.

***Index Terms*—** Visualization, Smoothing methods, Image generation, Filters, Image enhancement

1. INTRODUCTION

Over many years of experience, painters, sculptures, artists, researchers all realized how much effective the direction information is in visual perception. Together with perception, the ways of expressing direction information effectively has attracted a lot of attention lately.

One of the accurate and famous ways of visualizing direction information is Line Integral Convolution (LIC) [1]. LIC investigates the direction of a vector field locally along a well defined piecewise-linear curve. This property of LIC makes it accurate and reliable. While being suitable for many scientific applications (computer graphics, mechanics etc.), such an ability of being local makes it also very desirable for artistic applications.

In image processing, from a non photo realistic rendering (NPR) point of view, preservation of some saliency while performing a significant abstraction, can tweak the brain and create shape information while omitting the unneccesary details. Various painters such as Van Gogh, noticed this technique which is more decorative and less realistic, and used it in painting styles such as impressionism.

One can generate vector fields in various ways to extract certain features of the image, and use it in LIC to create different rendering styles. In fact, researchers tried many approaches to express different styles of digital art. E.g., in [2] a method using LIC for generating stylistic lines is suggested. Also, a method for generating multilayer pencil drawings [3] and a technique to imitate the impressionist effect on photographic images [4], both of which use LIC to visualize a vector field, are good examples of the researchers' approach.

We suggest a new method to obtain a vector field for LIC. This idea based on Wieckert's coherence enhancing diffusion filtering (CED) [5] which can extract the oriented flow structures in the image by utilizing the second moment matrix (structure tensor). When merged with LICs visualization power, the flow directions of an image are visualized smoothly.

This paper is organized as follows: First, we give and explain the formal definition of LIC. Then we present Wieckert's method for obtaining flow directions. The final section describes how flow directions can be used in LIC and presents results of our approach compared with other methods in literature.

2. LINE INTEGRAL CONVOLUTION (LIC)

Line integral convolution, which is a general method suggested by Cabral and Leedom [1], can be used to approximate and visualize two or three dimensional vector fields. It takes a vector field (V) and an input image (F) as input. An image and its intensity gradient can be used to generate a smoother image, or a noise image and intensity gradient of an image is used to obtain a texture image, as the vector field and image inputs, respectively [1].

LIC is a derivative of a method based on convolution algorithms described by Van Wijk[6] and Perlin[7] combined with Digital Differential Analyzer (DDA) line drawing techniques [8]. The main difference is that, LIC uses a local streamline to generate the filter instead of using a single DDA line. This streamline is formed by starting from a pixel location (x, y) and moving out by the forward coordinate advection equation given in (1):

$$P_0 = (x + 0.5, y + 0.5)$$

$$P_i = P_{i-1} + \frac{V(\lfloor P_{i-1} \rfloor)}{\|V(\lfloor P_{i-1} \rfloor)\|} \Delta s_{i-1} \tag{1}$$

$V(\lfloor P_i \rfloor) =$ the vector from V at lattice point $(\lfloor P_x \rfloor, \lfloor P_y \rfloor)$

$$s_e = \begin{cases} \infty & \text{if } (V \parallel e) \\ 0 & \text{if } (\frac{\lfloor P_c \rfloor - P_c}{V_c} < 0) \\ \frac{\lfloor P_c \rfloor - P_c}{V_c} & \text{otherwise} \end{cases} \tag{2}$$

$$\Delta s_i = \min(s_{top}, s_{bottom}, s_{left}, s_{right})$$

where $(e, c) \in \{(top, y),(bottom, y),(left, x),(right, x)\}$.

To obtain symmetry about the center of the streamline, it should also be advected in the reverse direction as shown in (3). The definitions of the variables are the same except the prime sign which only denotes that advection is in the negative direction.

$$P'_0 = P_0$$

$$P'_i = P'_{i-1} - \frac{V(\lfloor P'_{i-1} \rfloor)}{\|V(\lfloor P'_{i-1} \rfloor)\|} \Delta s'_{i-1} \tag{3}$$

(1) and (3) are iterated L times for a streamline composed of $2L$ line segments.For each segment i, an integral of a convolution kernel $k(w)$ (e.g. a box filter) is computed (4):

$$h_i = \int_{s_i}^{s_i+\Delta s_i} k(w)\, dw. \tag{4}$$

where $s_0 = 0$ and $s_i = s_{i-1} + \Delta s_{i-1}$. The value of output image F' at location (x, y) is calculated as follows:

$$F'(x, y) = \frac{\sum_{i=0}^{l} F(\lfloor P_i \rfloor)\, h_i + \sum_{i=0}^{l'} F(\lfloor P'_i \rfloor)\, h'_i}{\sum_{i=0}^{l} h_i + \sum_{i=0}^{l'} h'_i} \tag{5}$$

where $F(\lfloor P \rfloor)$ is the value of the pixel at $(\lfloor P_x \rfloor, \lfloor P_y \rfloor)$ and $l = j$ such that $s_i \leq L < s_{i+1}$.

The value of L strongly affects the result. For small L, F' becomes very similar to F; and for large L, all pixel values of F' becomes similar.

There are some situations at which singularities occur during LIC calculation. When the vector field has 0 value for any pixel location, or $s_i = s_{i+1}$ for any stream line segment; the singularity is handled by truncating the streamline at that point. If 0 value occurs at the center of streamline, an arbitrary value or a predetermined value can be used instead, in order to be able to calculate (5) at that point.

3. FLOW ENHANCING LINE INTEGRAL CONVOLUTION (FELIC)

The key point of our method is to identify the flow patterns in an image, and obtain a vector field showing in the flow directions. We then make use of this field to apply LIC either to the image for smoothing, or to white noise for texture generation. Identifying flow field is mainly composed of three parts: Computing the robust gradients, the structure tensor, and finally the normalized set of eigenvectors which will lead us to the flow field.

3.1. Structure Tensor

Structure tensor is a positive semidefinite, reliable derivative matrix which describes the local structure. Assume a very simple local structure descriptor ∇u_σ, as the gradient of a Gaussian smoothed image. Even though this is a well known edge detector, it's not a very appropriate smoother, since as the noise scale (σ) of the Gaussian gets small, the result would end up being fluctuated. As σ gets larger, the operator loses its ability to preserve information. This effect is mainly due to opposite signs in the neighborhood canceling each other. To overcome this problem, the gradient operator can simply be replaced by its tensor product $\nabla u_\sigma \otimes \nabla u_\sigma$, which is the structure tensor. The resulting matrix look like:

$$S = \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \tag{6}$$

In (6), I_x denotes the Gaussian derivative of u in x direction, and I_y being the same operator in y direction.

$$I_x = g_{x,\sigma} * u, \text{ and } I_y = g_{y,\sigma} * u \tag{7}$$

where:

$$g_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{8}$$

Because S matrix describes purely orientations (without incorporating the sign), a smoothing on the structure tensor will yield a smoothing only on the orientation field. It's well known that the orthanormal set of eigenvectors of this matrix describes the directions where local contrast is maximal [5], [9]; and the eigenvalues measure the magnitude of this contrast. Hence, the dominant eigenvector (normalized eigenvector corresponding to the maximum eigenvalue) describes the direction where the contrast is maximal. This is orthogonal to the flow direction.

3.2. Optimized Rotation Invariance (ORI) Kernels

Unlike classical gradient detectors such as Sobel, which make use of only 3x3 stencils, optimized rotation invariance kernels utilize 5x5 stencils, which are more robust, accurate and which can avoid blurring of artifacts. reference. ORI kernels were already used in diffusion filtering such as coherence enhancing diffusion [10]. In our algorithm, instead of approximating spatial derivatives using Sobel or simple central differencing, we benefit from ORI Kernels, which are defined as:

$$O_x = \begin{bmatrix} -3 & 0 & 3 \\ -10 & 0 & 10 \\ -3 & 0 & 3 \end{bmatrix} \text{ and } O_y = O_x^T \tag{9}$$

3.3. Flow Based Direction Field

The characteristics of the structure tensor, and its eigenvectors form a solid basis on the computation of the direction of flow. Because structure tensor is only 2x2, its orthanormal eigenvectors and eigenvalues can easily be computed analytically as following:

$$w_x = \frac{2I_xI_y}{\sqrt{(I_y^2 - I_x^2 + \sqrt{(I_x^2 - I_y^2)^2 + 4I_xI_y})^2 + 4I_xI_y}} \tag{10}$$

$$w_y = \frac{I_y^2 - I_x^2 + \sqrt{(I_x^2 - I_y^2)^2 + 4I_xI_y}}{\sqrt{(I_y^2 - I_x^2 + \sqrt{(I_x^2 - I_y^2)^2 + 4I_xI_y})^2 + 4I_xI_y}} \tag{11}$$

$$\mu_{1,2} = \frac{I_x^2 + I_y^2 \pm \sqrt{(I_x^2 - I_y^2)^2 + 4I_xI_y}}{2} \tag{12}$$

where $W = (w_x, w_y)$ is an eigenvector and W^T is orthagonal to W. Because $W \perp W^T$, when one eigenvector is computed, obtaining the value for the other is straightforward. μ_1 and μ_2 are the eigenvalues of W and W^T, respectively.

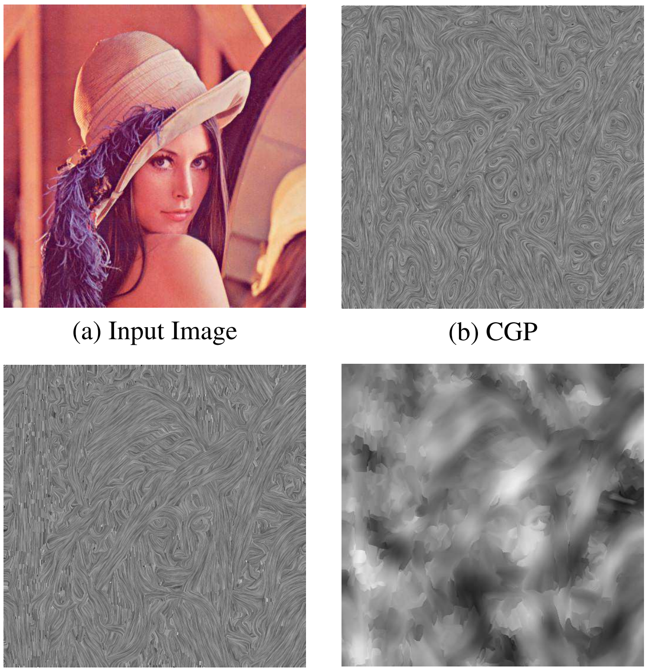
The dominant eigenvector corresponds to the direction with largest contrast. At this point, we sample, at every pixel, the non-dominating normalized eigenvector as our flow direction, forming the direction field. As the eigenvalues are a measure for constrast, this corresponds to the eigenvector with minimal eigenvalue. Such a method can be defined as:

$$V = \begin{cases} W, & \mu_1 < \mu_2 \\ W^T & \text{otherwise} \end{cases} \tag{13}$$

After performing (13), we end up with a vector field V. To perform FELIC, it sufficient to use V as input in classical LIC algorithm described in Section 2.

4. RESULTS

Firstly, various algorithms are used to generate texture images for an image given in Figure 1 (a). The result in (b) is obtained by using a method which uses continuous glass patterns (CGP) [11]. The results in (b) and (c) are obtained by white noise, whereas (d) is obtained by using perlin noise.



(a) Input Image (b) CGP

(d) FELIC with white noise (e) FELIC with perlin noise

Fig. 1. Textures generated with CGP and FELIC.

Secondly, various algorithms are used to smooth an image given in Figure 2 (a). The outputs of CED [5], LIC method [1] and FELIC method are presented in Figure 2 (b), (c) and (d), respectively. Again, intensity gradient is used as the input vector field of LIC.

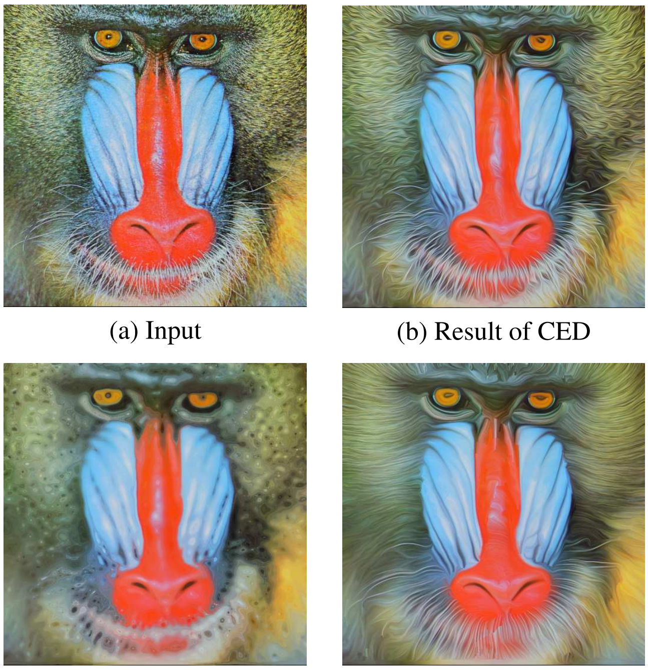
In addition, the effect of increasing L is presented in Figure 3. For larger L, smoothing in the direction of flow present in the leaves and the waterfall becomes much more remarkable.

As can be seen from the figures, FELIC is highly capable of capturing flow directions and representing them smoothly. By performing FELIC, it is also possible to smooth an image in an intelligent way.

5. CONCLUSION

We have introduced a new approach for texture generation and image smoothing by making use of LIC and flow directions. Our results show that the suggested method successfuly performs these operations. In terms of smoothing, it is visible that our operator outperforms its predecessor Coherence Enhancing Diffusion, and our suggestion is capable of producing highly representative textures.

The use of the suggested method is not limited with the context we dealed with in this paper. It is also possible to ex-



(a) Input (b) Result of CED

(c) Result of LIC (d) Result of FELIC

Fig. 2. Comparison of CED, LIC and FELIC Used for Smoothing.

tend the work by using FELIC as the basis of currently available NPR effects, such as [4], to improve them visually.

6. REFERENCES

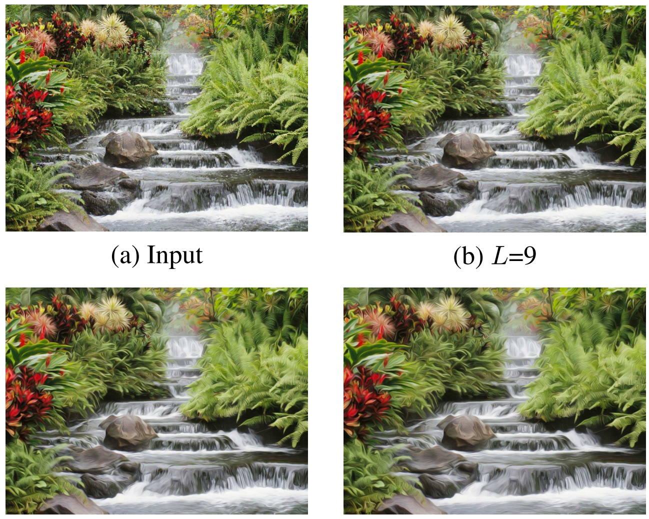
[1] Brian Cabral and Leith (Casey) Leedom, "Imaging vector fields using line integral convolution," in *Proceedings of Computer Graphics*. ACM SIGGRAPH, 1993, pp. 263–272.

[2] Henry Kang, Seungyong Lee, and Charles K. Chui, "Coherent line drawing," in *Proceedings of Symposium on Non-photorealistic Animation and Rendering*. ACM, 2007, pp. 43–50.

[3] Shigefumi Yamamoto, Xiaoyang Mao, and Atsumi Imamiya, "Enhanced lic pencil filter," in *Proceedings of International Conference on Computer Graphics, Imaging and Visualization*, 2004, pp. 251–256.

[4] Chung-Ming Wang and Jiunn-Shyan Lee, "Using ilic algorithm for an impressionist effect and stylized virtual environments," in *Journal of Visual Languages & Computing*, 2003, vol. 14, pp. 255–274.

[5] Joachim Weickert, "Coherence-enhancing diffusion filtering," in *International Journal of Computer Vision*, 1999, pp. 111–127.



(a) Input (b) L=9

(c) L=21 (d) L=33

Fig. 3. Outputs of FELIC for various L values.

[6] Jarke J. Van Wijk, "Spot noise: Texture synthesis for data visualization," in *Proceedings of Computer Graphics*. ACM SIGGRAPH, 1991, vol. 25, pp. 309–318.

[7] Ken Perlin, "An image synthesizer," in *Proceedings of Computer Graphics*. ACM SIGGRAPH, 1985, pp. 287–296.

[8] Jack E. Bresenham, "Algorithm for computer control of a digital plotter," in *IBM Systems Journal 4*, 1965, pp. 25–30.

[9] Joachim Weickert, "Coherence-enhancing shock filters," in *Pattern Recognition Lecture Notes in Computer Sciences*, 2003, vol. 2781, pp. 1–8.

[10] Joachim Weickert and Hanno Scharr, "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance," in *Prooceedings of J. Visual Communication and Image Representation*, 2002, vol. 13, pp. 103–118.

[11] Giuseppe Papari and Nicolai Petkov, "Continuous glass patterns for painterly rendering," in *Transactions on Image Processing*. IEEE, 2009, vol. 18, pp. 652–664.